Subscriptions Downloads Containers Support Cases

Products & Services Knowledgebase How to begin NFS Debugging

⚠ A translation for your language preference does not exist.

How to begin NFS Debugging

Updated 12 Novembre 2020 alle 10:02 - English ▾

Contents

There are a number of metrics and data that can be collected when an NFS problem has been identified. This document describes some of these common steps that can accelerate the discovery of the root cause, by breaking NFS issues down into four major categories and describing a troubleshooting process for each:

- Performance Issues (Reading or Writing data)
- Protocol Issues
- Connectivity Issues
- Kernel Crash

Performance Issues (Reading or Writing data)

Define a repeatable benchmark that runs for at least 15 minutes. Try using a subset of the actual NFS I/O workload as a benchmark. Capture the following statistics on the NFS client:

nfsiostat

Run `nfsiostat` to monitor the performance of NFS reads and writes:

```
date > nfsiostat.txt; nfsiostat 5 NFS-MOUNT-POINT >> nfsiostat.txt
```

The output will show the `ops/s`, `kB/s` and latencies for reads/writes, i.e. It describes the data travelling via the network to the NFS Server and how long it takes. For example, NFS write performance is poor, it looks like this is due to a network or NFS server latency because `avg RTT` should be much lower:

```
         write:  ops/s           kB/s          kB/op        retrans        avg RTT (ms)
avg exe (ms)
                14.000       11410.775       815.055      0 (0.0%)        1107.043
80432.143
```

Where:

```
"avg RTT": "the network + server latency of the request"
"avg exe": "the total time the request spent from init to release"
```

/proc/meminfo

When the NFS I/O involves `WRITE` operations, look at Dirty, Writeback and NFS Unstable values. You could capture something like this every 5 seconds to correspond to the `nfsiostat` capture times:

```
while true; do date >> nfs_meminfo.txt; cat /proc/meminfo | egrep "
(Dirty|Writeback|NFS_Unstable):" >> nfs_meminfo.txt; sleep 5; done
```

Where:

```
Dirty: Memory which is waiting to get written back to the disk
Writeback: Memory which is actively being written back to the disk
NFS_Unstable: NFS pages sent to the server, but not yet committed to stable storage
```

The Dirty count will increase when there's data to be written to a device. When a threshold is met it will start writing the dirty pages to a device and Writeback will increase accordingly. If write I/O ceases, then Dirty count will keep dropping as Writeback occurs. Eventually the Dirty count can drop to zero whilst there is still some Writeback left to be completed.

iostat

Capture CPU and IO device usage:

```
iostat -cxt 5 >> nfs_util.txt
```

tcpdump

On the NFS client, capture a representative sample of traffic for approx. 5 minutes between the NFS client and NFS server:

```
tcpdump -n -C 250M -s 300 -i INTERFACE -w OUTPUTFILE host NFS-SERVER-IP
```

The above tcpdump command only collects the first 300 bytes of every packet (`-s 300`).

The above tcpdump command "rolls over" into separate 250Mb files (`-C 250M`). To save uploading every one of the "rolling" output files, the timespan which each file covers can be determined with the `capinfos` tool from the `wireshark` package:

```
$ capinfos OUTPUTFILE.pcap.10 | grep time\:
Start time:          Wed Jul 16 17:25:21 2014
End time:            Wed Jul 16 17:26:28 2014
```

mountstats

This command is useful in obtaining NFS mount options, buffered versus direct IO, RTT latency per RPC procedure, backlog latency per RPC procedure.

- Copy the raw mountstats before:

```
cp /proc/self/mountstats  mountstats_raw_before_$(date +%s).txt
```

- Run the reproducer
- Copy the raw mountstats after:

```
cp /proc/self/mountstats  mountstats_raw_after_$(date +%s).txt
```

The mountstats command can provide you a summary of NFS activity between the two captures:

```
mountstats -f mountstats_raw_after_* -S mountstats_raw_before_*
```

SysRq-t

It is useful to capture the stack traces of the kernel processes to identify if and where processes are getting blocked. Please refer to How to use the SysRq facility to collect information from a RHEL server on how to enable `SysRq` .

Then do the following a few times, perhaps 3 times in 5 second intervals:

```
echo t > /proc/sysrq-trigger
```

Verifying available network bandwidth

The following information has been provided by Red Hat, but is outside the scope of the posted Service Level Agreements and support procedures. Installing unsupported packages does not necessarily make a system unsupportable by Red Hat Global Support Services; however, Red Hat Global Support Services will be unable to support or debug problems with packages not shipped in standard RHEL channels. Installing packages from EPEL is done at the user's own risk.

Install `iperf3` from EPEL.

- On the NFS server:

```
iperf3 -i 10 -s
```

- On the NFS client:

```
iperf3 -i 10 -w 4M -t 60 -c NFS-SERVER-IP
```

Refer to Using iperf to test network bandwidth throughput for more detailed instructions or alternative bandwidth measurement methods.

Repeatable Benchmark

If there is no repeatable benchmark, `dd` can be used to artificially generate a load:

Write

Remember to use `conv=fsync` for write benchmarks. e.g.

```
dd if=/dev/zero of=/mnt/nfs/file conv=fsync bs=1M count=1000
```

You can also use `oflag=direct` to compare cached I/O and direct I/O

Read

Remember to drop caches or unmount and mount the filesystem to ensure data is being read from the NFS server and not the NFS client's cache. e.g:

```
echo 3 > /proc/sys/vm/drop_caches
dd if=/mnt/nfs/file of=/dev/null bs=1M count=1000
```

A more detailed baseline collection is described at Initial baseline data collection for NFS client streaming I/O performance.

Protocol Issues

It is critical to identify the time/date when an error occurs. That helps us focus on a smaller period within the data provided to us. Do *NOT* expect someone to look through hundreds of MBs of data if a time/date is not provided.

When it is applicable, make sure you capture data both when the problem can be reproduced and when the problem can not be reproduced. That allows comparing the two sets of data to identify the differences.

Capture the following data on the NFS client:

tcpdump

When it is not a performance issue involving the transfer of a lot of Read or Write data, please make it a priority to capture a tcpdump:

```
tcpdump -n -C 250M -s 0 -i INTERFACE -w OUTPUTFILE host NFS-SERVER-IP
gzip OUTPUTFILE
```

strace

If applicable also capture an strace:

```
strace -T -tt -f -v -s 4096 [-o OUTPUTFILE] <command>
```

If using RHEL 6.7 or later, or RHEL 7 or later, add the `-yy` flags as well.

More detailed instructions are provided at How do I use strace to trace system calls made by a command?

mount

If you are having difficulties mounting an NFS export on the NFS client. Then run `mount` with the verbose option:

```
mount -vvv [USUAL-OPTIONS] NFS-SERVER:/EXPORT NFS-MOUNT-POINT
```

NFS user space daemons (ps -e | egrep 'rpc\.')

Read the respective `man` pages of these daemons.

Debugging/verbosity can be enabled by editing `/etc/sysconfig/nfs` like so:

```
RPCMOUNTDOPTS="-d all"
RPCIDMAPDARGS="-vvv"
RPCGSSDARGS="-vvv"
RPCSVCGSSDARGS="-vvv"
```

NOTE: `RPCMOUNTDOPTS` and `RPCSVCGSSDARGS` are applicable only to the NFS *Server*. `RPCGSSDARGS` is applicable only to the NFS *Client*.

To get debug output between `rpc.nfsd` and kernel `nfsd` :

```
RPCNFSDARGS="-d"
```

Ports

- Use `rpcinfo -p` to dump the list of RPC programs registered with the portmapper.
- Use 'rpcinfo -t/-u' to actually test connectivity with an RPC program running on a remote host. For example to test connectivity with mountd via UDP (which is the default for the MOUNT protocol) running on 'server.example.com':

```
# rpcinfo -u server.example.com 100005 3
program 100005 version 3 ready and waiting
```

- Inspect `/etc/sysconfig/nfs` to see if ports have been modified.
- Ensure portmapper is running on Port 111. Including with NFSv4.

NFS Server export options

- Provide the NFS export options of the NFS export on the NFS server. e.g. On a RHEL NFS Server:

```
cat /etc/exports
```

rpcdebug

The nfs/nfsd/sunrpc debugging output can be useful after looking at a tcpdump and having some idea what is going wrong.

Note: This debugging dumps a lot of data to syslog and can slow down older systems. Therefore, requesting this data from the customer is discouraged unless we are confident in what we are looking for.

- Enable debug:

```
rpcdebug -m nfs -s all
rpcdebug -m nfsd -s all
rpcdebug -m rpc -s all
rpcdebug -m nlm -s all
```

- Disable debug:

```
rpcdebug -m nfs -c all
rpcdebug -m nfsd -c all
rpcdebug -m rpc -c all
rpcdebug -m nlm -c all
```

Note: On an NFS server, the relevant modules are `nfsd` , `rpc` and `nlm` . On an NFS client, the relevant modules are `nfs` , `rpc` and `nlm` .

- To get a list of modules and valid flags, run:

```
rpcdebug -vh
```

Kerberos

- Please ensure that Kerberos is setup correctly before trying to use Kerberos with NFS. e.g: On RHEL 5, Configuring a Kerberos 5 Server.

Connectivity Issues

Check for dropped packets

It is very important to first check if there are problems with the NIC and the NIC driver. For example refer to RHEL network interface dropping packets.

tcpdump

Note: If the NFS Client is reporting the error, " `nfs: server [...] not responding` ", then use the `tcpdump-watch.sh` script attached below.

Collect tcpdump on both the NFS client and NFS server while reproducing the problem.

On the NFS client run:

```
tcpdump -n -C 250M -s 300 -i INTERFACE -w OUTPUTFILE host NFS-SERVER-IP
```

On the NFS server run:

```
tcpdump -n -C 250M -s 300 -i INTERFACE -w OUTPUTFILE host NFS-CLIENT-IP
```

The above tcpdump command only collects the first 300 bytes of every packet (`-s 300`).

The exception to this is when troubleshooting either NFSv4 issues, or `READDIR` / `READDIRPLUS` issues, or issues with batched NFS commands. In those cases, capture the full packet length with the `-s 0` option instead of `-s 300` .

The above tcpdump command "rolls over" into separate 250Mb files (`-C 250M`). To save uploading every one of the "rolling" output files, the timespan which each file covers can be determined with the `capinfos` tool from the `wireshark` package:

```
$ capinfos OUTPUTFILE.pcap.10 | grep time\:
Start time:          Wed Jul 16 17:25:21 2014
End time:            Wed Jul 16 17:26:28 2014
```

Try to map warnings/errors in syslog to packets in the tcpdump according to their timestamps.

SysRq-t

It is useful to capture the stack traces of the kernel processes to identify if and where processes are getting blocked. Please refer to How to use the SysRq facility to collect information from a RHEL server on how to enable `SysRq` .

Then do the following a few times, perhaps 3 times in 5 second intervals:

```
echo t > /proc/sysrq-trigger
```

Example

- RHEL mount hangs: nfs: server [...] not responding, still trying

---

Kernel Crash

If the RHEL NFS Client or NFS Server Linux kernel has crashed / panicked / oopsed, then please upload the vmcore file. If a vmcore file was not generated, please provide the oops message from syslog.

Attaccamento

tcpdump-watch.sh

Prodotto **Red Hat Enterprise Linux** **Categoria** **Troubleshoot**

Componente **kernel** **nfs-utils** **Tag** **nfs** **nfs2** **nfs3** **nfs4** **troubleshooting**

Tipo articolo **General**

People who viewed this article also viewed

Capture VLAN tags by using tcpdump

Solution - Dec 4, 2017

How to capture network packets with tcpdump?

Solution - Apr 12, 2022

Is there a simple way to gather a packet capture on an NFS client mount point?

Article - Jan 2, 2017

5 Comments

YJ

NEWBIE

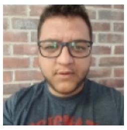
13 Points

28 June 2016 7:31 AM

yjeong-sa

Great article. :)

↩ Rispondere



EXPERT

PRO

841 Points

7 August 2017 9:13 AM

Diego Roberto Dos Santos

Amazing, thanks for sharing.

↩ Rispondere

MT

ACTIVE CONTRIBUTOR

130 Points

13 November 2017 10:33 PM

Matthew Taylor

This did not address the question of where is NFS activity logged under normal circumstances.

↩ Rispondere

HK

ACTIVE CONTRIBUTOR

164 Points

31 July 2019 11:44 PM

Hamid Khan

Good read

↩ Rispondere

DN

COMMUNITY MEMBER

31 October 2022 2:05 PM

Don Nutt

Thank you, perhaps I can find my phantom. The details are exactly what I need to troubleshoot.

45 Points

↩ Rispondere