

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-02/msg02761.html>

- *From:* Gianluca Alberici <gianluca@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 06 Feb 2008 22:55:02 +0100
-

Hi Chuck,

I finally got it. Problem and solution have been found from 6 month but nobody cared...up to now those servers have not been maintained, this problem is not discussed anywhere else than the following link. The bug (userspace server side i would say at this point) is well described from the author of an nfs-user-server patch which has not been managed yet. The magic hint to find it on google was 'nfs server utimensat' :-)

<http://marc.info/?l=linux-nfs&m=118724649406144&w=2>

I have already prepared a working patch for cfsd based upon the one ive listed. The nfs patch is of course waiting for commit since august, 2007. Ill submit it to debian cfsd mantainers, hoping to have more chance than my predecessor.

It doesnt seem to me that there was any kernel related issue.

Thanks a lot again, sorry for the lots of noise i have done. I will try to be more appropriate next time.

Cheers,

Gianluca

Chuck Lever wrote:

Hi Gianluca-

On Feb 6, 2008, at 1:25 PM, Gianluca Alberici wrote:

Hello all,

Thanks to Chuck's help i finally decided to proceed to a git bisect and found the bad patch. Is there anybody that has an idea why it breaks userspace nfs servers as we have seen ? Sorry for emailing directly Chuck Lever and Andrew Morton but i really wanted to thank Chuck for his precious help and thought that /akpm/ having signed this commit maybe he's going to figure out whats wrong easily

The commit you found is a plausible source of the trouble (based on our current theory about the problem).

What isn't quite clear to me is whether this commit causes your user-space server to start failing suddenly, or it causes the client to start sending the special non-standard time stamps in the SETATTR request. My guess is the latter, but I want to confirm this guess against reality :-)

Are you running the client and server concurrently on the same system? If so, it would be helpful if you could run this test with a constant kernel version on one side while varying it on the other. If client and server are already on different systems, can you tell us which system and which kernel combinations caused the failure?

A matrix of combinations might be:

1. server kernel is before 1c710c89, client kernel is before 1c710c89
2. server kernel is before 1c710c89, client kernel is after 1c710c89
3. server kernel is after 1c710c89, client kernel is before 1c710c89
4. server kernel is after 1c710c89, client kernel is after 1c710c89

Thanks.

This is what i finally get from git:

```
1c710c896eb461895d3c399e15bb5f20b39c9073 is first bad commit
commit 1c710c896eb461895d3c399e15bb5f20b39c9073
Author: Ulrich Drepper <drepper@xxxxxxxxxxx>
Date: Tue May 8 00:33:25 2007 -0700
```

utimensat implementation

Implement utimensat(2) which is an extension to futimesat(2) in that it

- a) supports nano-second resolution for the timestamps
- b) allows to selectively ignore the atime/mtime value
- c) allows to selectively use the current time for either atime or mtime
- d) supports changing the atime/mtime of a symlink itself along the lines of the BSD lutimes(3) functions

[...]

```
[akpm@xxxxxxxxxxxxxxxxxxxxx: add missing i386 syscall table entry]
Signed-off-by: Ulrich Drepper <drepper@xxxxxxxxxxx>
Cc: Alexey Dobriyan <adobriyan@xxxxxxxxxxx>
Cc: Michael Kerrisk <mtk-manpages@xxxxxxx>
Cc: <linux-arch@xxxxxxxxxxxxxxxxxxx>
Signed-off-by: Andrew Morton <akpm@xxxxxxxxxxxxxxxxxxxxx>
Signed-off-by: Linus Torvalds <torvalds@xxxxxxxxxxxxxxxxxxxxx>
```

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

```
:040000 040000 3bedbc7fd919ba167b8e5f208a630261570853bb
927002a9423dcb51ba4f7bee53e60cdca6c1df43 M arch
:040000 040000 fd688c5b534efd3111cbf1e1095d6ff631738325
3d0fbf20fb3da1cb380c92f5b2b39815897376d3 M fs
:040000 040000 bfb1a907a9a842db4fa3543e12a8381d4e11b1eb
9c1d99324db12e066c0d17870fe48457809ad43b M include
```

Thanks in advance, regards,

Gianluca

Hi Gianluca–

On Jan 30, 2008, at 7:40 AM, Gianluca Alberici wrote:

Hello again everybody

Here follows the testbench:

- I got two mirrors, same machine, same disk etc...changed hostname, IP, and on the second i have recompiled kernel.
- First: 2.6.21.7 on debian sarge
- Second: 2.6.22 same system.
- Onto both i got nfs–user–server and cfsd last versions
- The export file is the same (localhost /opt/nfs (rw, async), stripping off the async option does not changes anything)
- Mount options are exactly the same.

The problem arises in the very same manner with both nfs and cfsd:

```
NFS:setattr {
...
...
RPC:call_decode {
return 22;
}
...
return 22;
}
```

Again, there is nothing wrong with the RPC client or

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

call_decode. The *server* is returning NFSERR_INVAL (22) to a SETATTR request; the RPC client is simply passing that along to the NFS client, as it is designed to do.

I have tried these kernels:

- 2.6.16.11 works
- 2.6.20 works
- 2.6.21 works
- 2.6.21.7 works
- 2.6.22 doesnt work (contiguous to previous version)
- 2.6.23 doesnt work (same behavior as previous)
- 2.6.23.9 doesnt work (as above)
- 2.6.24rc7 doesnt work (as above)

I would really like to do more, client or server side, if you ave any suggestions. Can we find out what is the change (doesnt matter if it is a buf or bug fix) that caused this problem ?

The goal here is to identify the kernel change between 2.6.21 and 2.6.22 that makes the client generate SETATTR requests the user- space server chokes on. It may be a change in the NFS client, or it could be somewhere else in the file system stack, like the VFS.

The usual procedure is to use "git bisect". It does a binary search on the kernel patches between the working kernel version and the kernel version that is known not to work. It works like this:

1. You clone a linux kernel git repository (if you don't have a git repository already)
2. You tell git bisect which kernel version is working, and which isn't. git bisect then selects a commit about half way in between the working and non-working versions, and checks out that version of the kernel
3. You build that kernel, and run your test case

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

4. You tell git bisect whether the resulting kernel passes your test case, it selects a new commit, and checks out that version of the kernel.

5. Repeat steps 3 and 4 until git bisect has identified the commit that causes the kernel to stop passing your test case

If the number of patches between 2.6.21 and 2.6.22 is N , then git bisect will find the faulty patch in $O(\log_2(N))$ steps. For example, if there are 250 patches between 2.6.21 and 2.6.22, it will take about 8 iterations of steps 3 and 4 to find the faulty patch, if all goes well; far fewer than the total number of patches you would need to test one at a time.

Naturally you can also do this by applying and reverting patches with "patch -p1", but it's a little more work.

Chuck Lever wrote:

On Jan 29, 2008, at 3:31 PM, Trond Myklebust wrote:

On Tue,
2008-01-29
at 20:50
+0100,
Gianluca
Alberici
wrote:

Hello,

I
confirm
that
i
have
encountered
this
same
problem
(EINVAL
on
open

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

(...O
|
TRUNC)
with
the
following
userspace
servers:

–
nfs-user-server
shipped
with
debian
sarge/etch
etc...
–
cfsd
(crypto
file
system
which
is
an
nfs
server)

I
want
to
underline
again
that
these
userspace
servers
have
been
working
perfectly
until
2.6.21.7
(which
is
the
last
2.6.21)
Since
2.6.22
the
problem

came
out
and
it
is
still
present
into
2.6.24
rc7
(last
i
tested).
Conclusion:
there
must
have
been
something
that
is
changed
in
2.6.22
that
caused
the
problem.

The only
difference
between
these two
dumps are
the fact that
the first
one isn't
using the
Sun
convention
for telling
NFSv2
servers to
set to
the current
time (see
the code in

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

xdr_encode_current_server_time).

I thought I saw that on both
SETATTRs, but I could be
wrong.

I don't see
why this
would be
new
behaviour
after 2.6.21.
The code
for
this has
been in the
NFS client
since 2.6.15
at least...

A mount option is set on
one test client, and not the
other, perhaps?

--

Chuck Lever
chuck[dot]lever[at]oracle[dot]com

-

To unsubscribe from this
list: send the line
"unsubscribe linux-nfs" in
the body of a message to
majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at
<http://vger.kernel.org/majordomo-info.html>

-

Re: NFS EINVAL on open(... | O_TRUNC) on 2.6.23.9

To unsubscribe from this list: send the line
"unsubscribe linux-nfs" in
the body of a message to
majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at
<http://vger.kernel.org/majordomo-info.html>

--
Chuck Lever
chuck[dot]lever[at]oracle[dot]com

-
To unsubscribe from this list: send the line "unsubscribe
linux-nfs" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at
<http://vger.kernel.org/majordomo-info.html>

--
Chuck Lever
chuck[dot]lever[at]oracle[dot]com

-
To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>

--
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>