

DragonFlyBSD - Bug #1695

NFS-related system breakdown

03/13/2010 10:22 PM - Anonymous

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
Hi all.			
I do a cp(1) from /mnt/nfs (which is a zfs filesystem NFS-exported by an opensolaris installation) to my /home/beket directory (HAMMER fs).			
After a few MBs having being copied, cp(1) process stalls in 'getblk' state. Also it is unkillable by ^C. From that point there's downhill.			
Many commands will block if I issue them, such as 'mount' or 'df'. I can still view top(1) update its contents but the system is on the edge. dmesg shows sparse messages 'nfs server 10.0.0.1:/export/nfs: not responding' or '[diagnostic]: \$address block on cache_something ""'. If I break into the debugger, I get nothing unusual. scgetc < sckbdevent < kbd_xxxx < taskqueue_yyyy etc.			
I tried to kill X and the system hang to the point that a cold reset was necessary. I once managed to resume after typing 'c' in db> prompt, but that doesn't always succeed.			
This situation is, I think, 100% reproducible. Also, I don't have problems copying stuff from to a linux NFS client.			
Cheers			

History

#1 - 03/13/2010 10:33 PM - Anonymous

New data.

I hit ctrl+alt+delete and system stopped at 'unmount: there are still XXX namecache references'. Since it stayed there for a few minutes, I hit again ctrl+alt+delete and I got a kernel trap in nsfm_dissect. fault virtual address is 0xdeadc0ea :)

Stack trace is nsfm_dissect < nfs_request < nsfsvc_iod_reader < lwkt_exit.

I think I got a kernel core dump. Will upload to leaf tomorrow.

#2 - 03/13/2010 10:48 PM - Anonymous

Dunno if this helps to bisect the offending code, but forcing mount_nfs to use v2 protocol makes the problem go completely away. I have already copied the video like 10 times without encountering a b/lock situation.

If I unmount the filesystem and remount it as v3 this time, the problem reappears.

#3 - 03/14/2010 01:17 AM - dillon

You could try a NFSv3 UDP mount instead of the default TCP mount to see if that helps. If it doesn't then it should be possible to use tcpdump to monitor the nfs traffic and figure out which rpc is stalling.

You can also mount with the 'intr' option which will make blocked accesses interruptable.

There isn't enough information to determine who is at fault.

-Matt
Matthew Dillon
<dillon@backplane.com>

#4 - 03/14/2010 08:08 PM - Anonymous

If I do NFSv3 UDP it works fine. I copied the file 5x with no b/lock failure. I then remounted it as NFSv3 TCP and it failed as yesterday. I killed X and looked at dmesg. I now saw some new messages:

```
EXDEV case 1 0xd1b06a88
EXDEV case 1 0xd1b06a88
EXDEV case 1 0xd1b06a88
EXDEV case 1 0xd1b06a88
[diagnostic] cache_lock: blocked on $addr ""
[diagnostic] cache_lock: blocked on $addr ""
```

Oh, and yes -i(ntr) did the trick. I was able to ^C the cp(1) and restore normal operation.

What's next? tcpdump on the server side ?

#5 - 03/15/2010 09:30 AM - dillon

:
:What's next? tcpdump on the server side ?
:

I'm not sure tcpdump can track rpcs over a tcp connection, but that would be the next step to try to find out which side is responsible for dropping one of the RPCs on the floor.

Another possibility is that it is a stalled TCP connection which you can detect by looking at the receive and transmit buffer backlog on both sides via 'netstat -p tcp'.

-Matt
Matthew Dillon
<dillon@backplane.com>

#6 - 04/10/2014 12:35 AM - casusbubble

- *Description updated*

If I do NFSv3 UDP it works fine. I copied the file 5x with no b/lock failure.

<http://www.reverse-your-diabetes-today.net/>