

DragonFlyBSD - Bug #731

system freeze on "slice too large"

07/15/2007 01:11 PM - corecode

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:	tuxillo	% Done:	0%
Category:	Kernel	Estimated time:	0.00 hour
Target version:	3.8.0		
Description			
<p>hey,</p> <p>i've now had twice a nasty freeze (kind of) with something like this (hand transcribed):</p> <pre>dscheck(#ad/0x20021): slice too large 2/2 ..</pre> <p>then vinum tells me that it put "build" down and continues:</p> <pre>fatal: build.p0.s0 read error, offset 33831591936 for 4096 bytes build.p0.s0: user buffer offset 10209280000 for 4096 bytes</pre> <p>(more slice too large follow)</p> <p>then, the namecache does</p> <pre>blocked on 0xd4fb7b58 "corecode"</pre> <p>and repeats it every 30 seconds or so. system is unoperable at this point.</p> <p>breaking to the debugger works, but dumphsys does not work:</p> <pre>dumping to dev #ad/0x20023, blockno 2130432 dump failed, reason: area improper</pre> <p>i'm running:</p> <pre>DragonFly sweatshorts.home.corecode.ath.cx 1.9.0-DEVELOPMENT DragonFly 1.9.0-DEVELOPMENT #14: Sun Jun 17 11:03:58 CEST 2007 corecode@sweatshorts.home.corecode.ath.cx:/usr/build/obj/usr/build/src/sys/SWEATSHORTS i386</pre> <p>i've attached relevant outputs.</p> <p>thanks, simon</p>			

History

#1 - 07/15/2007 05:19 PM - c.turner

Simon 'corecode' Schubert wrote:

I'm definitely under the impression that vinum is unhappy with the new labels.

2 things perhaps worth discussion:

- your 'a' partition overlaps the vinum - I assume this means you're using a vinum root?
- was your volume set created prior to upgrading past the disklable changes?
- is this an occasional crash, or a constant one?

just chiming in to get some more data points out on this I suppose..
b/c this is most definably over my kernel-n00b head :)

test findings here from last night :

- able to create a new disk set (see the other thread)
- unable to use the disk set after reboot without a vinum resetconfig, reload config files kind of thing (e.g. vinum 'start' messes up and needs to be reminded of the disk layout)
- having 'vinum_enable="YES"' in rc.conf keeps me from booting up, as it appears to make the kernel very confused about it's various mount points (e.g. after vinum start, it 'forget's about existing 'raw partition' mounts and can't find getty, etc...)

when I booted with no vinum_enable and 'vinum list' yields no drives, and 'vinum start'ed causes the 'raw partition amnesia' to set in, I get some of those dscheck messages like you saw. I then couldn't drop to the debugger, and cntrl-alt-del resulted in some very interesting infinite loop printouts ..

perhaps the mountpoint confusion explains why the dump didn't work for you?

unfortunately, I don't have the problem machine hooked up to a serial console yet to snag the output, and due to other events I'm going to be unable to test for the next few weeks..

#2 - 07/15/2007 07:05 PM - dillon

I'm getting close to just throwing vinum away. It makes so many bad assumptions about EVERYTHING I'm getting seriously pissed at it.

vinum's automatic disk search is basically broken. It is making ridiculous assumptions about disk layout instead of doing what it SHOULD be doing, which is to scan available media for partitions marked as being type 'vinum'.

Try specifying an actual partition being used as a vinum drive and don't use the auto-start feature. When I do that, e.g. when I do 'vinum read /dev/vn0s0d' on my test disk, it finds the saved configuration.

When I try to do 'vinum start' vinum blows up my system. Literally blows it up. It is clearly leaving memory structures associated with the kernel that it has freed. vinum is such a huge mess I just don't have any idea where the problem is occurring, but I think it may be related to how vinum tries to scan available disk media.

So far the only 'slice too large' errors I've gotten occur when I try to do 'vinum start'. I think it is related to vinum's insistence on rolling its own everything. It constructs device numbers right out of the blue and then tries to use them.

-Matt
Matthew Dillon
<dillon@backplane.com>

#3 - 07/15/2007 07:23 PM - dillon

```
:hey,  
:  
:i've now had twice a nasty freeze (kind of) with something like this (hand transcribed):  
:  
:dscheck(#ad/0x20021): slice too large 2/2  
:..  
:  
:then vinum tells me that it put "build" down and continues:  
:....
```

Same thing. Don't use vinum's autostart feature. Try specifying an actual partition being used as a vinum drive.

I don't know regarding the read errors. That sounds like a different issue.

-Matt

#4 - 07/15/2007 07:30 PM - corecode

This is not a new kernel; vinum worked for at least 3 years here. The message do not appear on bootup: I am using root on vinum, so I have to specify the drives it should use.

Wouldn't dscheck return an error which makes a physical read to fail? That's probable why vinum gets a read error and shuts down the system.

cheers
simon

#5 - 07/15/2007 07:34 PM - corecode

This is an old kernel, no new labels here.

yes.

it was created about 3 years ago, I think.

it happened two times, within 12 hours or so.

cheers
simon

#6 - 07/15/2007 07:48 PM - dillon

:This is an old kernel, no new labels here.

:
:....

:yes.

:
:> - was your volume set created prior to upgrading past the disklabel
:> changes?

:
:it was created about 3 years ago, I think.

:
:> - is this an occasional crash, or a constant one?

:
:it happened two times, within 12 hours or so.

:
:cheers
: simon

I think there are two different issues here. This issue:

:i've now had twice a nasty freeze (kind of) with something like this (hand trans
:cribed):
:
:dscheck(#ad/0x20021): slice too large 2/2
:...

Is simply vinum trying to access e.g. ad0s2 when ad0 is dangerously
dedicated (i.e. only ad0s0 exists). When does it do this? This only
happens to me when I try to have vinum auto-start.

:then vinum tells me that it put "build" down and continues:
:
:fatal: build.p0.s0 read error, offset 33831591936 for 4096 bytes
:build.p0.s0: user buffer offset 10209280000 for 4096 bytes
:
:(more slice too large follow)
:
:then, the namecache does
:
:blocked on 0xd4fb7b58 "corecode"

Read error sounds like a real read error. Is that a valid offset?
It's a byte offset, not a sector number. Is there any disk redundancy?

I'm not sure what the two offsets are. They must be relative to
different things. The first is probably relative to the vinum volume
while the scnd looks like it may be partition-relative.

A real read error can cause UFS to get confused and create a namecache
blockage.

-Matt

#7 - 07/16/2007 11:02 AM - c.turner

Matthew Dillon wrote:

I didn't see that reaction coming :)

Confirming this works for a striped (raid0) setup using:

```
'vinum read /dev/ad1 /dev/ad3'
```

Using particular slices / partitions did not work.
This is the same configuration as before, with ad[13]s1a as a 'vinum' partition across the whole disk, I tried '/dev/ad[13]s1' and '/dev/ad[13]s1a'.

I'd initially just read one disk in at a time, leaving the stripe in 'flaky' state.

This persisted across reboots until I did a

```
'vinum setdaemon 1; vinum setstate up ...; vinum saveconfig'
```

to get things back, which held held across reboots as well, so saving the configuration, etc back to the disk seems ok. as I mentioned my schedule will prevent testing for a while - my apologies. Current state 'works for me' for the time being since I don't have a vinum root machine.

I've not gotten a chance to read the code yet.. do you see any other problems with it, aside from the disk assumptions & general 'bit rot' / need for TLC?

For some reason I like it, although perhaps I just like saying the word 'vinum' .. hmm.

#8 - 07/19/2007 08:59 PM - c.turner

Did some more investigating and it looks like the 'vinum start' essentially calls 'vinum read' after a devstat(3) iteration to find 'suitable devices', which based on a quick test program appear to be the actual disk devices (e.g. adN / sdN, etc). The difference seems to be that 'vinum read' with no arguments seems to probe things differently than 'vinum read' without arguments .. so I'll keep digging.. Perhaps a short term fix related to this might 'hold things' for the time being until the 'direct partition' scan you mention can be implemented.

also, there is some kgetenv related to vinum.disks checked during the load, so perhaps what you'd mentioned in the other thread about adding this for boot-time isn't needed .. ? Haven't tested yet / not sure if this was from your earlier patch or not - I still need to dig further.. will be attempting to do so over the next few days, hopefully.

#9 - 09/08/2009 09:03 PM - alexh

There have been a lot of changes relating to vinum and disk system (including dscheck). This probably isn't relevant anymore, but I'd appreciate someone confirming it.

Cheers,
Alex Hornung

#10 - 02/20/2014 07:46 AM - tuxillo

- Description updated
- Category set to Kernel
- Assignee changed from 0 to tuxillo
- Target version set to 3.8.0

Grab.

#11 - 02/20/2014 07:51 AM - corecode

- File signature.asc added

Is vinum still supported?

On 02/20/2014 04:46 PM, bugtracker-admin@leaf.dragonflybsd.org wrote:

> Issue [#731](#) has been updated by tuxillo.

>

> Description updated

> Category set to Kernel

> Assignee changed from 0 to tuxillo

> Target version set to 3.8.0

>

> Grab.

>

> -----

> Bug [#731](#): system freeze on "slice too large"

> <http://bugs.dragonflybsd.org/issues/731#change-11804>

>

> * Author: corecode

> * Status: New

> * Priority: Normal

> * Assignee: tuxillo

> * Category: Kernel

> * Target version: 3.8.0

> -----

> hey,

>

> i've now had twice a nasty freeze (kind of) with something like this (hand transcribed):

>

> dscheck(#ad/0x20021): slice too large 2/2

> ..

>

> then vinum tells me that it put "build" down and continues:

>

> fatal: build.p0.s0 read error, offset 33831591936 for 4096 bytes

> build.p0.s0: user buffer offset 10209280000 for 4096 bytes

>

> (more slice too large follow)

>

> then, the namecache does

>

> blocked on 0xd4fb7b58 "corecode"

>

> and repeats it every 30 seconds or so. system is unoperable at this point.

>

> breaking to the debugger works, but dumphsys does not work:

>

> dumping to dev #ad/0x20023, blockno 2130432

> dump failed, reason: area improper

>

> i'm running:

> DragonFly sweatshorts.home.corecode.ath.cx 1.9.0-DEVELOPMENT DragonFly 1.9.0-DEVELOPMENT [#14](#): Sun Jun 17 11:03:58 CEST 2007

> corecode@sweatshorts.home.corecode.ath.cx:/usr/build/obj/usr/build/src/sys/SWEATSHORTS i386

>

> i've attached relevant outputs.

>

> thanks,

> simon

>

> ---Files-----

> disklabel-ad2s1.txt (527 Bytes)

> disklabel-ad4s1.txt (577 Bytes)

> fdisk-ad2.txt (741 Bytes)

> fdisk-ad4.txt (735 Bytes)

> vinum-l.txt (3 KB)

>

>

#12 - 02/20/2014 08:02 AM - tuxillo

- File deleted (signature.asc)

Files

disklabel-ad2s1.txt	527 Bytes	07/15/2007	corecode
disklabel-ad4s1.txt	577 Bytes	07/15/2007	corecode
fdisk-ad2.txt	741 Bytes	07/15/2007	corecode
fdisk-ad4.txt	735 Bytes	07/15/2007	corecode

