

DragonFlyBSD - Bug #1583

panic: assertion: cursor->trans->sync_lock_refs > 0 in hammer_recover_cursor

10/22/2009 06:46 PM - corecode

Status:	In Progress	Start date:	
Priority:	Normal	Due date:	
Assignee:	tuxillo	% Done:	0%
Category:	VFS subsystem	Estimated time:	0.00 hour
Target version:	3.8.0		
Description			
Happened when I started 8 mirror-stream at the same time.			
(kgdb) bt			
#0 _get_mycpu () at ./machine/thread.h:83			
#1 dumpsys () at /usr/src/sys/kern/kern_shutdown.c:657			
#2 0xc019ba89 in boot (howto=260) at /usr/src/sys/kern/kern_shutdown.c:378			
#3 0xc019bdf5 in panic (fmt=0xc032b90c "assertion: %s in %s") at /usr/src/sys/kern/kern_shutdown.c:813			
#4 0xc0299c7f in hammer_recover_cursor (cursor=0xe84826b4) at /usr/src/sys/vfs/hammer/hammer_cursor.c:591			
#5 0xc02a5139 in hammer_ioc_mirror_write (trans=0xe8482a50, ip=0xea5ce6d0, mirror=0xe51a6f88) at /usr/src/sys/vfs/hammer/hammer_mirror.c:465			
#6 0xc02a37b8 in hammer_ioctl (ip=0xea5ce6d0, com=3234097160, data=0xe51a6f88 "", fflag=1, cred=0xc6787308) at /usr/src/sys/vfs/hammer/hammer_ioctl.c:142			
#7 0xc02b65e4 in hammer_vop_ioctl (ap=0xe8482aac) at /usr/src/sys/vfs/hammer/hammer_vnops.c:2305			
#8 0xc0203107 in vop_ioctl (ops=0xe4795b70, vp=0xea5f22e8, command=3234097160, data=0xe51a6f88 "", fflag=1, cred=0xc6787308, msg=0xe8482cf0) at /usr/src/sys/kern/vfs_vopops.c:389			
#9 0xc0202b75 in vn_ioctl (fp=0xe4585910, com=3234097160, data=0xe51a6f88 "", ucred=0xc6787308, msg=0xe8482cf0) at /usr/src/sys/kern/vfs_vnops.c:938			
#10 0xc01c3c1b in fo_ioctl (fd=3, com=3234097160, uspc_data=0xbfbff954 <Address 0xbfbff954 out of bounds>, map=0x0, msg=0xe8482cf0) at /usr/src/sys/sys/file2.h:88			
#11 mapped_ioctl (fd=3, com=3234097160, uspc_data=0xbfbff954 <Address 0xbfbff954 out of bounds>, map=0x0, msg=0xe8482cf0) at /usr/src/sys/kern/sys_generic.c:697			
#12 0xc01c3ccb in sys_ioctl (uap=0xe8482cf0) at /usr/src/sys/kern/sys_generic.c:521			
#13 0xc03088ea in syscall2 (frame=0xe8482d40) at /usr/src/sys/platform/pc32/i386/trap.c:1339			
#14 0xc02f3356 in Xint0x80_syscall () at /usr/src/sys/platform/pc32/i386/exception.s:876			
kernel+core nr. 13 on leaf:~corecode/crash			
cheers simon			

History

#1 - 10/23/2009 12:50 AM - dillon

:Happened when I started 8 mirror-stream at the same time.

:

:cheers

: simon

Yours are easier to track down :-). Try the patch below.

Tomokazu's are more difficult.

-Matt
Matthew Dillon
<dillon@backplane.com>

```
diff --git a/sys/vfs/hammer/hammer_mirror.c b/sys/vfs/hammer/hammer_mirror.c
index 574b74d..05e12a0 100644
--- a/sys/vfs/hammer/hammer_mirror.c
+++ b/sys/vfs/hammer/hammer_mirror.c
@@ -462,8 +462,10 @@ hammer_ioc_mirror_write(hammer_transaction_t trans, hammer_inode_t ip,
 */
if (error == EDEADLK) {
while (error == EDEADLK) {
+ hammer_sync_lock_sh(trans);
hammer_recover_cursor(&cursor);
error = hammer_cursor_upgrade(&cursor);
+ hammer_sync_unlock(trans);
}
} else {
if (error == EALREADY)
```

#2 - 01/30/2010 06:10 PM - floid

Contents of patch applied to 2.4, no ill effects noted. With the PFSes mostly-synchronized I didn't have enough data left to sync to test for the same failure with multiple `mirror-copy`s running, but I'll be setting up parallel mirror-streams shortly as required for my setup. (Is there a convenient way to destroy a PFS and make sure it gets recreated with the same PFS number?)

I briefly thought the low I/O figures on da1 (mirroring from da0 to da1) might have been a new quirk:

```
# hammer mirror-copy /DATA/ /Mirror/pfs/DATA
histogram range 0000000107e975cf - 000000010d6514f0
Mirror-read: Mirror from 0000000000000002 to 000000010d6514f0
Mirror-read /DATA/ succeeded
```

```
> iostat -w1
tty      ad0      da0      da1      cpu
tin tout KB/t tps MB/s KB/t tps MB/s KB/t tps MB/s us ni sy in id
2 465 0.00 0 0.00 0.00 0 0.00 0.00 0 0.00 4 0 10 1 85
0 80 0.00 0 0.00 64.00 643 40.21 64.00 4 0.25 21 0 42 2 35
0 80 0.00 0 0.00 63.82 683 42.59 60.00 4 0.23 28 0 37 3 32
0 80 0.00 0 0.00 63.79 614 38.26 64.00 6 0.37 20 0 38 1 41
```

...but it completed in appropriate time and all the data appears to be there, so I'm not sure if HAMMER was being smart enough to skip over what existed from a previous interrupted copy or what. I don't really remember whether the pre-patch stats actually looked any different for the destination disk.

[SATA disks on a mpt0, write caches enabled.]

#3 - 01/30/2010 06:34 PM - dillon

:I briefly thought the low I/O figures on da1 (mirroring from da0 to da1) might :have been a new quirk:

```
:
:> iostat -w1
:  tty      ad0      da0      da1      cpu
: tin tout KB/t tps MB/s KB/t tps MB/s KB/t tps MB/s us ni sy in id
: 2 465 0.00 0 0.00 0.00 0 0.00 0.00 0 0.00 4 0 10 1 85
: 0 80 0.00 0 0.00 64.00 643 40.21 64.00 4 0.25 21 0 42 2 35
: 0 80 0.00 0 0.00 63.82 683 42.59 60.00 4 0.23 28 0 37 3 32
: 0 80 0.00 0 0.00 63.79 614 38.26 64.00 6 0.37 20 0 38 1 41
:
```

...but it completed in appropriate time and all the data appears to be there, so :I'm not sure if HAMMER was being smart enough to skip over what existed from a :previous interrupted copy or what. I don't really remember whether the :pre-patch stats actually looked any different for the destination disk.

:
:[SATA disks on a mpt0, write caches enabled.]

The more likely case is that the mirror stream repeated the data but since the target already had it there was no need to re-write it on the target. In that case the only disk activity on the target will be accessing and comparing the meta-data.

It tries to stage the mirroring stream so it can pick up where it left off but it's a bit of a hack. If it managed to complete a batch then it certainly will pick up after the completion and not resend the whole batch. I need to redo the code to actually calculate the batch size and split things up based on that instead of splitting things up based on B-Tree key ranges.

This applies to initial mirroring only, of course. Once the initial copy is finished all later mirroring runs will be incremental.

-Matt
Matthew Dillon
<dillon@backplane.com>

#4 - 02/20/2014 07:31 AM - tuxillo

- *Description updated*
- *Category set to VFS subsystem*
- *Status changed from New to In Progress*
- *Assignee changed from 0 to tuxillo*
- *Target version set to 3.8.0*

Grab.