DragonFlyBSD - Bug #1336

Still looking for reports of missed directory entries w/ HAMMER

04/14/2009 04:49 PM - dillon

Status:	In Progress	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	VFS subsystem	**Estimated time:**	0.00 hour
Target version:	4.2.x		

Description

I am still looking for any reports of missed directory entries w/
HAMMER, from 'ls', 'cpdup', or any other program. So far I have
not been able to reproduce the problem, but at least two people have
reported a similar issue in the last few months.

-Matt
Matthew Dillon
<dillon@backplane.com>

History

#1 - 04/15/2009 07:28 AM - ftigeot

I may have found a similar problem.

I run rsnapshot on a dedicated hammer filesystem, mounted nohistory.
Recently, I have discovered some directories couldn't be properly rotated:

```
# du -sh *
du: weekly.0/anego.zefyris.com/usr/local/share/icons/hicolor/128x128/stock/text:
No such file or directory
0B    weekly.0
du: weekly.2/anego.zefyris.com/home/ftigeot/Maildir/.Dragonfly-kernel/cur: No such file or directory

# ls -la stock
ls: text: No such file or directory
total 0
drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 .
drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 ..

# rmdir stock
rmdir: stock: Directory not empty

# rm -r stock
rm: stock/text: No such file or directory
rm: stock: Directory not empty
```

Some directory entries are not visible with the normal tools, yet seem to
exist someway.

The machine crashed violently recently (power failure); the hammer filesystem
was re-mounted without any apparent error.

#2 - 04/16/2009 07:01 AM - dillon

```
:I may have found a similar problem.
:
:I run rsnapshot on a dedicated hammer filesystem, mounted nohistory.
:Recently, I have discovered some directories couldn't be properly rotated:
:
:# du -sh *
:du: weekly.0/anego.zefyris.com/usr/local/share/icons/hicolor/128x128/stock/text:
:No such file or directory
:  0B    weekly.0
:du: weekly.2/anego.zefyris.com/home/ftigeot/Maildir/.Dragonfly-kernel/cur: No such file or directory
:
:# ls -la stock
:ls: text: No such file or directory
```

```
:total 0
:drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 .
:drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 ..
:
:# rmdir stock
:rmdir: stock: Directory not empty
```

Is there any chance that those particular files or directories were
being actively modified when the crash occured? Or would they have
been stable at the time of the crash?

That looks like a case where the directory entry exists but the inode
does not. I have seen this occur before in crash recovery cases but
I had thought I had fixed it. There's was an edge case where a directory
entry can get synced to disk in a different transaction then the inode.
If the machine crashes right then you wind up with the above situation.

If the files should have been stable then try rebooting the machine
and see if the problem is still present. That will tell me whether
its a namecache effect in the kernel or something that got synced
to the media.

I think this may be a different issue then the ls/cpdup problem,
though the more I think about it the more a namecache issue makes
sense w/ regards to the ls/cpdup problem.

-Matt
Matthew Dillon
<dillon@backplane.com>

#3 - 04/16/2009 10:45 AM - ftigeot

On Wed, Apr 15, 2009 at 11:57:21PM -0700, Matthew Dillon wrote:

```
>
> :I run rsnapshot on a dedicated hammer filesystem, mounted nohistory.
> :Recently, I have discovered some directories couldn't be properly rotated:
> :
> :# ls -la stock
> :ls: text: No such file or directory
> :total 0
> :drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 .
> :drwxr-xr-x  1 root  wheel     0B Apr  2 04:03 ..
> :
> :# rmdir stock
> :rmdir: stock: Directory not empty
>
>    Is there any chance that those particular files or directories were
>    being actively modified when the crash occured?  Or would they have
>    been stable at the time of the crash?
```

I can't say for sure, but the probability is high the machine crashed during a
recopy operation.
This can take a long time (too many files, 28K directories per run)

```
>    That looks like a case where the directory entry exists but the inode
>    does not.  I have seen this occur before in crash recovery cases but
>    I had thought I had fixed it.  There's was an edge case where a directory
>    entry can get synced to disk in a different transaction then the inode.
>    If the machine crashes right then you wind up with the above situation.
```

The buggy directories were created after I upgraded to DragonFly-2.2.

```
>    If the files should have been stable then try rebooting the machine
>    and see if the problem is still present.  That will tell me whether
>    its a namecache effect in the kernel or something that got synced
>    to the media.
```

The media is definitely corrupt: I rebooted the machine and it is still
impossible to delete the directories.
All error messages stay the same.

Additionally, the following kernel messages were emitted during the last two days
(after the reboot):

```
Warning: BTREE_REMOVE: Defering parent removal2 @ 80000013e5b2d000, skipping
```

Warning: BTREE_REMOVE: Defering parent removal2 @ 8000002ba5b63000, skipping
Warning: BTREE_REMOVE: Defering parent removal2 @ 800000369defc000, skipping
Warning: BTREE_REMOVE: Defering parent removal2 @ 800000372a107000, skipping

#4 - 04/16/2009 04:23 PM - dillon

:I can't say for sure, but the probability is high the machine crashed during a
:recopy operation.
:This can take a long time (too many files, 28K directories per run)
:
:>   That looks like a case where the directory entry exists but the inode
:>   does not. I have seen this occur before in crash recovery cases but
:>   I had thought I had fixed it. There's was an edge case where a directory
:>   entry can get synced to disk in a different transaction then the inode.
:>   If the machine crashes right then you wind up with the above situation.
:
:The buggy directories were created after I upgraded to DragonFly-2.2.
:
:>   If the files should have been stable then try rebooting the machine
:>   and see if the problem is still present. That will tell me whether
:>   its a namecache effect in the kernel or something that got synced
:>   to the media.
:
:The media is definitely corrupt: I rebooted the machine and it is still
:impossible to delete the directories.
:All error messages stay the same.
:
:Additionally, the following kernel messages were emitted during the last two days
:(after the reboot):
:
:Warning: BTREE_REMOVE: Defering parent removal2 @ 80000013e5b2d000, skipping
:Warning: BTREE_REMOVE: Defering parent removal2 @ 8000002ba5b63000, skipping
:Warning: BTREE_REMOVE: Defering parent removal2 @ 800000369defc000, skipping
:Warning: BTREE_REMOVE: Defering parent removal2 @ 800000372a107000, skipping
:
:--
:Francois Tigeot

Those warnings can be ignored. It just means HAMMER couldn't immediately
remove an empty B-Tree leaf because someone else had one of the
parent B-Tree nodes locked. In fact, I will remove the message from
the sources.

I think the media issue is probably due to the crash. It isn't actually
corrupt, i.e. the UNDO works properly, but the directory entry wound
up getting created in a different transaction then the inode and the
crash occured inbetween, so we wound up with a directory entry and no
inode post-crash.

I will adjust HAMMER to allow those dead entries to be removed and look
into the flush sequencing again, after I've tracked down the directory
issue.

Dealing with link counts on inodes is actually quite difficult
because there can be multiple directory entries pending in different
transactions. I'm very careful to sync the inode's link count after
discounting directory entries queued for later flush transactions.
However, I think from your report there must be a bug where the initial
creation of an inode is not occuring in the same flusn transaction as
the creation of the related directory entry.

--

On the 'ls' issue.. its definitely a different issue. I got another
report from Peter who saw it happen on Avalon. It's definitely some
sort of transient cache effect or a bug in the run-time that does
NOT effect the media. I'm going to try to track that one down first.
Maybe its a bug in HAMMER's getdirentries() routine.

-Matt
Matthew Dillon
<dillon@backplane.com>

#5 - 04/17/2009 09:16 PM - dillon

Ok, here is what I've come up with so far:

* ls sometimes appears to not list some entries, but then they show up.

- The case where the effect is temporary, ls appears not to report
an error and there's somewhat of a question-mark as to why.

- The case where the effect is permanent, and ls reports errors for
some files, is a different issue related to crash recoery. I
will be making it possible to delete such files, but I want to
track down the temporary case first.

* zsh has a correction feature. For zsh the temporary issue appears as
a request to correct the name of a file, but then offer exactly
the same name as the correction. Peter Avalos hit this with e.g.
vi <somefilename> sometimes.

- Zsh does an access() call, which apparently fails, and then
reads the contents of the directory to come up with the proposed
correction... the directory appears to contain the correct
filename.

* cpdup. I have seen this temporary issue with cpdup from NFS to HAMMER.

- cpdup does not report an error if it creates a sub-directory as
part of the cpdup operation, but is then unable to stat or chown
it. This causes cpdup to incorrectly believe that the target
directory is in a different filesystem and it silently fails to
copy anything into it.

The bug is line 840 in /usr/bin/cpdup.c. I will commit a fix
right now so it reports the error.

- But when I check manually with ls the target directory DOES exist,
but cpdup failed to copy anything into it.

From this information it is my belief that this issue is due to a
cache effect when a file or directory is created or recently created.
The effect can cause lookups of the file or directory to fail, even
though the creation actually succeeded. When the cache effect goes
away the entry becomes visible on the media.

The directory appears to properly contain the entry, but it cannot be
stat()'d etc. But when the cache effect in the kernel self-corrects
(probably due to simply being discarded), the entry can be stat()'d
again. This is different from the permanent media issue reported by
Francois Tigeot.

So now I am trying to find a case where newly created or modified or
renamed files somehow get confused in the kernel cache. That's where
I am. No smoking gun yet but a lot of clues.

-Matt

#6 - 04/19/2009 01:09 AM - neil

Hi Matt,

First, thanks for Hammer. This is an awesome piece of work.

I normally use NetBSD, but was drawn to install Dragonfly on a new
machine to try out Hammer. I installed yesterday for the first
time. I think I hit this directory entry thing twice yesterday,
the first time I dismissed it as "I must have done something wrong",
but the second time I went slower and it was unmistakeable. I've
just found this thread on the net; let me describe what I did
which is perhaps a little unusual.

I checked /usr/pkgsrc out of cvs. When I want to install a package;
I can rarely remember which category a package is in, I do:

```
$ cvs /usr/pkgsrc
$ cd */mplayer     [I think this wildcard is key; bash is the shell]
$ sudo bmake install
```

mplayer was the second time I had this happen to me; I cannot remember
which package I was trying to install first time.

Now the bmake command failed with "don't know how to make install".
Sure enough, the directory had only two files in it and neither
was a Makefile so thinking something went wrong with my original
cvs checkout the I did "sudo cvs up". I can't remember what the
two files were (but I think they were the same both times this
happened), but because cvs didn't fail, CVS must have been a
subdirectory. cvs was quiet; nothing updated. So thinking something
was corrupted I did "cd .." "rm -r mplayer" "sudo cvs up". And
again, all commands succeeded quietly, cvs did nothing, and
the mplayer directory I had just removed wasn't recreated.

This is when I noticed I was in "/usr/pkgsrc/audio". I went back
to /usr/pkgsrc and did "cd */mplayer" and this time I was in
/usr/pkgsrc/multimedia/mplayer, which had a fully intact and usable
directory. The first "cd */mplayer" had taken me to a non-existent
place called "/usr/pkgsrc/audio/mplayer" with just one or two files
in it.

I hope something is useful to salvage from this, and sorry it's a
bit vague. I've only had an issue in interactive use. i.e. of
the 250 packages I build yesterday, bmake never got confused by
the filesystem like this, but then it doesn't use wildcards. Apart
from my own fumbling with /usr/pkgsrc above, there has been nothing
else amiss about the FS or the machine.

Neil.

#7 - 04/19/2009 07:09 AM - ftigeot

On Thu, Apr 16, 2009 at 09:18:24AM -0700, Matthew Dillon wrote:
> :I can't say for sure, but the probability is high the machine crashed during a
> :recopy operation.
> :This can take a long time (too many files, 28K directories per run)
> :
> :>   That looks like a case where the directory entry exists but the inode
> :>   does not. I have seen this occur before in crash recovery cases but
> :>   I had thought I had fixed it. There's was an edge case where a directory
> :>   entry can get synced to disk in a different transaction then the inode.
> :>   If the machine crashes right then you wind up with the above situation.
> :
> :The media is definitely corrupt: I rebooted the machine and it is still
> :impossible to delete the directories.
> :All error messages stay the same.
>
>   I think the media issue is probably due to the crash. It isn't actually
>   corrupt, i.e. the UNDO works properly, but the directory entry wound
>   up getting created in a different transaction then the inode and the
>   crash occured inbetween, so we wound up with a directory entry and no
>   inode post-crash.

I have more information: a new corrupt directory has appeared this morning and
the machine had *not* crashed (uptime: 5 days).

For some reason, rsnapshot seems to be really good at triggering this sort of
bug.

#8 - 04/19/2009 09:58 AM - neil

Please ignore this report. It's nothing to do with Hammer; I just
experienced the same thing under a NetBSD kernel after installing
NetBSD on a 2nd partition. It seems to be caused somehow by CVS's
lameness in creating and then removing all old directories during
checkout.

Neil.

#9 - 04/19/2009 08:27 PM - dillon

:
:Please ignore this report. It's nothing to do with Hammer; I just
:experienced the same thing under a NetBSD kernel after installing
:NetBSD on a 2nd partition. It seems to be caused somehow by CVS's

:lameness in creating and then removing all old directories during
:checkout.
:
:Neil.

Ah Shucks! I was hoping we'd found a way to reliably reproduce
it.

-Matt
Matthew Dillon
<dillon@backplane.com>

#10 - 04/20/2009 05:06 AM - dillon

:I have more information: a new corrupt directory has appeared this morning and
:the machine had *not* crashed (uptime: 5 days).
:
:For some reason, rsnapshot seems to be really good at triggering this sort of
:bug.
:
:--
:Francois Tigeot

It might actually be related. If we're getting an intermittent
file-not-found a file-create which occurs at the same time could
break something.

-Matt
Matthew Dillon
<dillon@backplane.com>

#11 - 04/21/2009 05:00 PM - dillon

Here's another question: How many rsnapshots or rsyncs are running
on the destination machine at the same time? Are you backing up
all your other machines and partitions in parallel to the target box?

-Matt

#12 - 04/21/2009 10:06 PM - ftigeot

On Tue, Apr 21, 2009 at 09:56:47AM -0700, Matthew Dillon wrote:

> Here's another question: How many rsnapshots or rsyncs are running
> on the destination machine at the same time? Are you backing up
> all your other machines and partitions in parallel to the target box?

This machine is pretty much dedicated to backup and file serving.

- it runs rsnapshot every 4 hours on the problematic filesystem.
Each rsnapshot run gets files in sequence from 7 different machines and 34
directory trees.
The target is a 400GB Hammer filesystem on a RAID1 volume (3Ware SATA). This
filesystem is mounted nohistory.
Rsnapshot takes a _long_ time to rotate its hourly directory snapshots
(directory deletion and re-creation, hard-links). My guess would be about 20
minutes per run for this part.

Time spent running rsync is quite short in comparison; this filesystem is only
running at most one instance of rsync.

- every 5 hours it runs a different rsync sequence and gets files from 6
machines and 28 remote directories
The target filesystem is a 250GB Hammer volume on a stand-alone SATA disk.
There is no special directory rotation, directories are always synchronized in
the same place; Hammer snapshots are used for history.
I have no trouble with this filesystem.

Apart from the backup activities, this machine exports some others (different
from the 2 backup fs above) filesystem with NFS. They are lightly used, and
then mainly for reading.

#13 - 04/22/2009 10:53 PM - dillon

Ok, I still haven't managed to trigger either issue but I've done
considerable reformulating of HAMMER's merged search code and

added some kprintf()'s to try to solve particular situations.

There is a risk in using this patch, so beware of that. I haven't
blown up my test box with it but neither was I able to hit any of
the kprintf()'s or related code paths that try to solve what I
believe the issue to be.

Perhaps you can cause some of these special situations to occur and
produce kprintf output and/or test to see if further issues arise
(or new issues, or the machine panics from something in the patch).

fetch http://apollo.backplane.com/DFlyMisc/hammer01.patch

-Matt

#14 - 04/23/2009 01:31 AM - qhwt+dfly

On Wed, Apr 22, 2009 at 03:49:37PM -0700, Matthew Dillon wrote:
> fetch http://apollo.backplane.com/DFlyMisc/hammer01.patch

Matt, shouldn't the following chunk (in above patch) be tested/pushed
independently of the rest of the patch?

```
@@ -2308,9 +2336,10 @@ hammer_sync_record_callback(hammer_record_t record, void *data)
* Assign the create_tid for new records. Deletions already
* have the record's entire key properly set up.
*/
- if (record->type != HAMMER_MEM_RECORD_DEL)
+ if (record->type != HAMMER_MEM_RECORD_DEL) {
record->leaf.base.create_tid = trans->tid;
record->leaf.create_ts = trans->time32;
+ }
for (;;) {
error = hammer_ip_sync_record_cursor(cursor, record);
if (error != EDEADLK)
```

#15 - 04/23/2009 03:23 AM - dillon

:YONETANI Tomokazu <qhwt+dfly@les.ath.cx> added the comment:
:
:Matt, shouldn't the following chunk (in above patch) be tested/pushed
:independently of the rest of the patch?

Nah. The create_ts was supposed to be set along with create_tid.
create_ts is not used by anything other then the undo code anyway
so nothing bad can happen.

-Matt
Matthew Dillon
<dillon@backplane.com>

```
:- if (record->type !=3D HAMMER_MEM_RECORD_DEL)
:+ if (record->type !=3D HAMMER_MEM_RECORD_DEL) {
:   record->leaf.base.create_tid =3D trans->tid;
:   record->leaf.create_ts =3D trans->time32;
:+ }
```

#16 - 04/23/2009 03:24 AM - dillon

:YONETANI Tomokazu <qhwt+dfly@les.ath.cx> added the comment:
:
:Matt, shouldn't the following chunk (in above patch) be tested/pushed
:independently of the rest of the patch?

Nah. The create_ts was supposed to be set along with create_tid.
create_ts is not used by anything other then the undo code anyway
so nothing bad can happen.

-Matt
Matthew Dillon
<dillon@backplane.com>

```
:- if (record->type !=3D HAMMER_MEM_RECORD_DEL)
:+ if (record->type !=3D HAMMER_MEM_RECORD_DEL) {
:   record->leaf.base.create_tid =3D trans->tid;
:   record->leaf.create_ts =3D trans->time32;
:+ }
```

#17 - 04/29/2009 11:06 PM - dillon

:This machine is pretty much dedicated to backup and file serving.
:
:- it runs rsnapshot every 4 hours on the problematic filesystem.
:Each rsnapshot run gets files in sequence from 7 different machines and 34
:directory trees.
:The target is a 400GB Hammer filesystem on a RAID1 volume (3Ware SATA). This
:filesystem is mounted nohistory.
:Rsnapshot takes a _long_ time to rotate its hourly directory snapshots
:(directory deletion and re-creation, hard-links). My guess would be about 20
:minutes per run for this part.
:
:Time spent running rsync is quite short in comparison; this filesystem is only
:running at most one instance of rsync.
:...
:Francois Tigeot

Ah ha. Ok, I will focus on rsnapshot's directory snapshot rotation
in trying to reproduce this issue.

Note: Just today I committed work to HEAD which allows messed up
directory entries to be removed. They will show up as the fifo
type in an ls (as well as spew warnings on the console), and you
can 'rm' them.

-Matt
Matthew Dillon
<dillon@backplane.com>

#18 - 01/14/2015 05:08 PM - tuxillo

- Description updated

- Category set to VFS subsystem

- Status changed from New to In Progress

- Assignee deleted (0)

- Target version set to 4.2.x

Hi guys,

What's the status of this?

Cheers,
Antonio Huete