fs - Bug #14254

failed pjd chown test 117

01/06/2016 05:59 AM - Greg Farnum

Status:	Resolved	**Start date:**	01/06/2016
Priority:	High	**Due date:**	
Assignee:	Zheng Yan	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Source:	other	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:		**Component(FS):**	
Regression:	No	**Labels (FS):**	
Severity:	3 - minor	**Pull request ID:**	
Reviewed:			

Description

http://pulpito.ceph.com/gregf-2016-01-04_11:47:54-fs-master---basic-mira/13222/

```
2016-01-04T13:28:49.073 INFO:tasks.workunit.client.0.mira105.stdout:Test Summary Report
2016-01-04T13:28:49.074 INFO:tasks.workunit.client.0.mira105.stdout:-------------------
2016-01-04T13:28:49.074 INFO:tasks.workunit.client.0.mira105.stdout:../pjd-fstest-20090130-RC/test
s/chown/00.t   (Wstat: 0 Tests: 171 Failed: 1)
2016-01-04T13:28:49.075 INFO:tasks.workunit.client.0.mira105.stdout:  Failed test:  117
2016-01-04T13:28:49.075 INFO:tasks.workunit.client.0.mira105.stdout:Files=191, Tests=1964, 377 wal
lclock secs ( 2.58 usr  1.86 sys +  4.54 cusr  8.47 csys = 17.45 CPU)
2016-01-04T13:28:49.076 INFO:tasks.workunit.client.0.mira105.stdout:Result: FAIL
2016-01-04T13:28:49.076 INFO:tasks.workunit:Stopping ['suites/pjd.sh'] on client.0...
```

Associated revisions

Revision dafb46ba - 01/15/2016 05:19 AM - Yan, Zheng

mds: delay handling client caps until corresponding inode is created

When handling client caps in clientreplay stage, it's possible that
corresponding inode does not exist because the client request which
creates inode hasn't been replayed. To handle this corner case, we
delay handling caps message until corresponding inode is created.

Fixes: #14254
Signed-off-by: Yan, Zheng <zyan@redhat.com>

History

#1 - 01/07/2016 01:14 AM - Greg Farnum

Okay, I believe the important sequence is:

- client creates inode, mds replies unsafe
- client requests inode change gid 65533, mds replies unsafe
- client requests inode change gid 65532, mds crashes
- mds goes into replay
- client sends off caps, which include inode on gid 65533
- mds does not recognize inode number
- mds drops client cap update on clientreplay_start, because it can't find inode

- client replays create and two setattrs
- client does not update trace on replies because it has already has same seq (1)
- client sends cap update, which still has gid 65533, mds accepts it

So I'm not quite sure what to do here. We have tids set by the client, and we have seqs set by the server, but seqs get reset when the MDS does and cap tids aren't related to regular op tids. I think for any missing inodes we ought to be trying to apply the caps after the messages get replayed, rather than just dropping them, but that actually doesn't help us here because we'd be going backwards anyway.

...actually, I don't see what keeps a client cap flush from overwriting a setattr change under non-failure cases, if for some reason the response manages to take long enough for caps to get flushed in the middle.

#2 - 01/07/2016 01:29 AM - Greg Farnum

Okay, this is running into code from the uid/gid enforcement stuff. From https://github.com/ceph/ceph/commit/1957aeddbf05f2ecf3be0a760ff5b5c313370eea in particular, which is justified by the comment in https://github.com/ceph/ceph/commit/ac031443 — but I don't think that's a true statement. Many of our other update functions invoke check_caps, but _setattr doesn't and even if it did (prior to making changes) there's no reason to think the dirtied cap bits would force an immediate flush. So I think we need to force a flush if we're doing a sync setattr while we have dirty caps.

But that's not a complete fix. We still need a way of ordering cap updates with respect to requested setattr operations.

#3 - 01/07/2016 01:36 AM - Greg Farnum

- Status changed from New to Verified

https://github.com/ceph/ceph/pull/7136 addresses the need to flush

#4 - 01/07/2016 02:05 AM - Greg Farnum

- Priority changed from Urgent to High

Okay, so the only way we can get into this trouble is if:
1) the inode isn't found prior to replay (ie, it got created but not journaled)
2) we have **competing** local dirty cap state and outstanding setattr requests.

And we can't get those competing states outside of the weird allowed-by-the-capabilities-but-not-our-security state case we're looking at now. So we just need to handle that.

#5 - 01/08/2016 02:42 PM - Zheng Yan

- Assignee set to Zheng Yan

#6 - 01/11/2016 09:39 AM - Zheng Yan

I interpret this differently.

- client creates inode, mds replies unsafe
- client requests inode change gid 65533, client marks Ax dirty
- client requests inode change gid 65532, client sends setattr request to MDS
- mds crashes and goes into reconnect stage

- client re-sends the setattr request (client does not got unsafe, see Client::resend_unsafe_requests)
- mds decide not to handle the setattr request in client_replay stage. (see Server::dispatch)
- client re-sends the cap flush with gid 65533 (see Client::early_kick_flushing_caps)
- mds goes to client_replay stags
- mds drops client cap update on clientreplay_start, because it can't find inode
- mds replays the create
- mds goto to active
- mds handles the setattr request and send replay to client (
- client re-sends the cap flush with gid 65533 again (see Client::kick_flushing_caps)
- client does not update trace on replies because it has already has same seq (1)
- client sends cap update, which still has gid 65533, mds accepts it

#7 - 01/12/2016 09:19 AM - Zheng Yan

- Status changed from Verified to Need Review

https://github.com/ceph/ceph/pull/7199

#8 - 01/20/2016 02:13 AM - Greg Farnum

- Status changed from Need Review to Resolved