fs - Bug #24802

races with nfs-ganesha reboots and delegation handling

07/06/2018 06:07 PM - Jeff Layton

Status:	New	**Start date:**	07/06/2018
Priority:	Normal	**Due date:**	
Assignee:	Jeff Layton	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	v15.0.0		
Source:		**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:		**Component(FS):**	Client, Ganesha FSAL, libcephfs
Regression:	No	**Labels (FS):**	
Severity:	3 - minor	**Pull request ID:**	
Reviewed:			

Description

So I've come up with a thought experiment that I think could be problematic for ganesha with delegations enabled. This scenario assumes current behavior where we do not "drain off" in-progress NFS RPCs before reporting that the grace period is being enforced:

```
Ganesha 1                Ganesha 2
---------                ---------
get delegation
                         block trying to get caps covered by delegation (either in client or on MDS
)
crash and restart
                         start enforcing grace, set enforcing flag
startup proceeds
kill off old state
                         blocked operation proceeds since caps are now freed
```

The last bit (blocked operation proceeding) occurs while a client of server 1 still technically holds a delegation. Which is a violation of basic tenets of this stuff.

We could fix this by "draining off" in progress RPCs before we report that we're enforcing, but that would just result in a deadlock in this situation. Ganesha-1 would be stuck at startup and never release its state. I think we probably do want to implement some sort of call draining like that, but we need to resolve the potential for deadlock here too.

What we'd like to happen here is for ganesha to return a retryable error on operations that will be blocked waiting on a delegation to be returned (i.e. NFS4ERR_GRACE or NFS4ERR_DELAY). The client can then wait a bit and redrive the thing. We don't have support for this in ceph though, and we'd need it.

One idea: define a new set of operations (or set some field in the session) that says that we won't block for "too long" waiting on caps. If we can't get the caps, give up and carry on elsewhere. That would probably fix cases where the Ceph client is blocked on a CapGet, but the MDS can also end up blocked gathering caps. I'm not sure what we can do there.

Thoughts?

History

#1 - 07/09/2018 01:48 PM - Patrick Donnelly

- Assignee set to Jason Dillaman

- Target version set to v14.0.0

#2 - 07/09/2018 01:50 PM - Jeff Layton

The right fix here is probably to push forward with end-to-end ceph client reclaim. The main problem is that there is a small window where the caps are not held which allows Ganesha 2 to race in and grab them. If we instead granted all of the state that the previous Ganesha 1 instance held before, then there would be no race window (Ganesha 2 would just continue blocking).

#3 - 07/09/2018 01:54 PM - Jeff Layton

- Assignee changed from Jason Dillaman to Jeff Layton

#4 - 03/07/2019 11:21 PM - Patrick Donnelly

- Target version changed from v14.0.0 to v15.0.0