CephFS - Bug #38326

mds: evict stale client when one of its write caps are stolen

02/14/2019 11:04 PM - Patrick Donnelly

Status:	Resolved	**% Done:**	0%
Priority:	Urgent		
Assignee:	Zheng Yan		
Category:	Correctness/Safety		
Target version:	v15.0.0		
Source:	Development	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	nautilus,mimic	**Component(FS):**	MDS
Regression:	No	**Labels (FS):**	task(hard)
Severity:	3 - minor	**Pull request ID:**	26737
Reviewed:		**Crash signature:**	

Description

IIUC: After mdsmap.session_time, the current behavior is that a stale session's caps' issued set is revoked and changed to CEPH_CAP_PIN. The MDS allows that stale session to later come back and "resume" by updating its cap.want set, which causes it to obtain new caps in the normal fashion.

One issue with this is that a client may be writing to a file, becomes unresponsive, and another client successfully begins (buffered) writing to that file concurrently. The only correct thing to do when another client comes along wanting that write cap is to evict the unresponsive client. Eviction is absolutely necessary because (a) we don't know when or if the client is coming back and (b) if the client is still connected to RADOS and writing bytes to the file while unable to receive/process messages from the MDS.

I would tentatively propose that the new behavior should be for a stale session:

(a) mark the session stale and check whether any locks are blocked by the newly "stale" caps. Ideally, we shouldn't invalidate a cap unnecessarily. (Why would we want to? Then the client needs to get the cap reissued which is expensive?)
(b) if a client comes along trying to obtain a conflicting WR/BUFFER/EXCL cap, evict the stale session immediately, wait for the osdmap update, then issue the cap.

If a stale session comes back, we can reissue most caps it already had because no other session has stolen its write caps. An exception is CEPH_CAP_GCACHE which may have been lost by an intervening write by another client.

Simple reproducer of the original problem with two clients:

1. [client 1] mkdir foo && pv -L 1K < /dev/urandom > foo/bar
2. kill -STOP <client1>
3. [client 2] pv -L 1K < /dev/urandom > foo/bar # client 2 blocks for 60s then gets write caps!
4. kill -CONT <client1> # both writes continue without buffer cap

Related issues:

Copied to CephFS - Backport #40326: nautilus: mds: evict stale client when on...	**Resolved**
Copied to CephFS - Backport #40327: mimic: mds: evict stale client when one o...	**Resolved**

History

#1 - 02/14/2019 11:06 PM - Patrick Donnelly

- Description updated

#2 - 03/04/2019 01:03 PM - Zheng Yan

- Status changed from 12 to Fix Under Review

- Pull request ID set to 26737

#3 - 03/07/2019 11:22 PM - Patrick Donnelly

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

#4 - 03/07/2019 11:27 PM - Patrick Donnelly

- Subject changed from mds: evict stale client when one of its write caps are stole to mds: evict stale client when one of its write caps are stolen

- Priority changed from Normal to Urgent

- Backport set to nautilus,mimic

#5 - 06/12/2019 09:07 PM - Patrick Donnelly

- Status changed from Fix Under Review to Pending Backport

Zheng, any issues backporting this?

#6 - 06/13/2019 10:26 AM - Nathan Cutler

- Copied to Backport #40326: nautilus: mds: evict stale client when one of its write caps are stolen added

#7 - 06/13/2019 10:26 AM - Nathan Cutler

- Copied to Backport #40327: mimic: mds: evict stale client when one of its write caps are stolen added

#8 - 06/19/2019 06:51 AM - Zheng Yan

- Status changed from Pending Backport to Fix Under Review

increment patches https://github.com/ceph/ceph/pull/28642

#9 - 07/01/2019 09:57 PM - Patrick Donnelly

- Status changed from Fix Under Review to Pending Backport

#10 - 10/23/2019 08:15 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved

While running with --resolve-parent, the script "backport-create-issue" noticed that all backports of this issue are in status "Resolved" or "Rejected".