Linux kernel client - Bug #38224

Race handling ceph_snap_context in kernel client

02/07/2019 02:59 PM - Luis Henriques

Status:	Fix Under Review	**% Done:**	0%
Priority:	Normal	**Spent time:**	0.00 hour
Assignee:	Zheng Yan		
Category:	fs/ceph		
Target version:			
Source:	Development	**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**ceph-qa-suite:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	

Description

I've been seeing generic/013 (from the xfstest suite) failing occasionally with a kmemleak warning. I finally found some time to look at it and... I can't say how many hours I've already spent looking, but I can say it's an embarassingly high number!

Here's the warning:

```
unreferenced object 0xffff8881fccca940 (size 32):
  comm "kworker/0:1", pid 12, jiffies 4295005883 (age 130.648s)
  hex dump (first 32 bytes):
    01 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<00000000d741a1ea>] build_snap_context+0x5b/0x2a0
    [<0000000021a00533>] rebuild_snap_realms+0x27/0x90
    [<00000000ac538600>] rebuild_snap_realms+0x42/0x90
    [<000000000e955fac>] ceph_update_snap_trace+0x2ee/0x610
    [<00000000a9550416>] ceph_handle_snap+0x317/0x5f3
    [<00000000fc287b83>] dispatch+0x362/0x176c
    [<00000000a312c741>] ceph_con_workfn+0x9ce/0x2cf0
    [<000000004168e3a9>] process_one_work+0x1d4/0x400
    [<000000002188e9e7>] worker_thread+0x2d/0x3c0
    [<00000000b593e4b3>] kthread+0x112/0x130
    [<00000000a8587dca>] ret_from_fork+0x35/0x40
    [<00000000ba1c9c1d>] 0xffffffffffffffff
```

After adding some debug code (tracepoints to collect some data), I found that it's a struct ceph_snap_context being leaked. Basically, after being created (which sets its initial refcount to 1), it's recount is being increased in ceph_queue_cap_snap():

```
...
update_snapc:
        if (ci->i_head_snapc) {
                ci->i_head_snapc = ceph_get_snap_context(new_snapc);
                dout(" new snapc is %p\n", new_snapc);
        }
        spin_unlock(&ci->i_ceph_lock);
...
```

The only ceph_put_snap_context for that ceph_snap_context object only occurs when umounting the filesystem, which leaves the refcount to 1, not freeing it's memory.

There's obviously a race *somewhere* and I suspect that either ci->i_head_snapc or ci->i_snap_realm are being set to NULL before a ceph_put_snap_context is done on that object. But the code is really quite complex, difficult to follow and I'm really out of ideas on how to debug this, specially because it's really difficult to reproduce the bug.

As anyone seen this issue before? Does anyone have any idea on how to proceed?

History

#1 - 02/08/2019 08:10 PM - Patrick Donnelly

- Project changed from CephFS to Linux kernel client

- Category set to fs/ceph

- Assignee set to Zheng Yan

- Start date deleted (02/07/2019)

- Source set to Development

Thanks for the report Luis!

#2 - 06/18/2019 07:33 AM - Zheng Yan

- Status changed from New to 7

fixed by "ceph: fix ci->i_head_snapc leak" in testing branch

#3 - 12/05/2019 09:44 PM - Patrick Donnelly

- Status changed from 7 to Fix Under Review