Linux kernel client - Bug #11960

Kernel panic when deleting a pool, which contains a mapped RBD

06/11/2015 10:04 AM - Alex Leake

Status:	Closed	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Ilya Dryomov		
Category:			
Target version:			
Source:	Community (dev)	**Reviewed:**	
Tags:	rbd, panic, delete	**Affected Versions:**	v0.94.2
Backport:		**ceph-qa-suite:**	rbd
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	

Description

Hello all,

I've found a reproducible bug in Ceph relating to mapped RBD images.

Steps to reproduce:

1. Create pool
2. Create RBD
4. Mount RBD
5. Write data to mount
6. Delete pool
7. Stop OSD cluster daemons (stop ceph-all)
8. Server which has RBD mapped will panic

We can reproduce this reliably. Here is our dev environment:

- Ubuntu 14.04 LTS, 3.16.0-38-generic (linux-generic-lts-utopic kernel)
- Ceph 0.94.1-1trusty packages
- megaraid_sas kernel module (default) on OSDs

I realise this is not a good idea to delete a pool, without first removing the RBD images - but, deleting them can take time. Also, I'm not always 100% sure if they are / are not mapped. The default behaviour should be to warn the user (dmesg?) - assuming this is a bug...

Kind Regards,
Alex.

Related issues:

Related to Linux kernel client - Bug #8568: libceph: kernel BUG at net/ceph/o...	**Closed**	**06/10/2014**
Related to Linux kernel client - Feature #9779: libceph: sync up with objecter	**Resolved**	**10/14/2014**

History

#1 - 06/11/2015 09:42 PM - Loïc Dachary

- Target version deleted (v0.94.2)

#2 - 06/11/2015 09:57 PM - Sage Weil

- Project changed from Ceph to Linux kernel client

- Category deleted (librbd)

- Assignee set to Ilya Dryomov

- Priority changed from Normal to Urgent

#3 - 06/12/2015 09:34 AM - Ilya Dryomov

- Status changed from New to Need More Info

Hi Alex,

While deleting a pool with mapped rbd images is just asking for trouble, we shouldn't panic. Can you provide some details on the panic (backtrace, etc) and your steps to reproduce in a form of a shell script? I tried but it didn't crash here.

#4 - 06/12/2015 05:14 PM - Alex Leake

Ilya,

Thanks for getting back to me. I'm away for the weekend, but I can get back to you on Monday.

Have a good weekend.

Alex.

#5 - 06/16/2015 01:44 PM - Alex Leake

- File after_removal.jpg added

- File panic.jpg added

Hello Ilya,

I've managed to re-create the issue, to make life easier I'll just explain our setup a bit.

We have three monitor servers, and three additional servers which contain 28 OSDs each. Each of the servers containing OSDs are called pyrum, prashadi, and pricei.

The script is in two sections.

If I run this on one of the OSD servers:

```
ceph osd pool create temp 4096
rbd create temp/test_rbd --size 1024
rbd map temp/test_rbd
parted -a optimal /dev/rbd0 mklabel gpt mkpart primary 0% 100%
mkfs -t xfs /dev/rbd0p1
mkdir /mnt/rbd
mount /dev/rbd0p1 /mnt/rbd
dd if=/dev/zero of=/mnt/rbd/data bs=512 count=2048
```

Then this on one of the monitors:

```
ceph osd pool delete temp temp --yes-i-really-really-mean-it
for host in pyrum prashadi pricei; do ssh $host stop ceph-all; done
```

Doing this reliably results in a panic on the server with the image mapped. I've attached two screenshots also. The first is just after deleting the pool (after_removal.jpg) the second is the actual panic (panic.jpg).

Let me know if you'd like any more info.

Regards,
Alex.

#6 - 06/22/2015 10:45 AM - Ilya Dryomov

- Status changed from Need More Info to Closed

Your panic.jpg shows only a tail of a panic splat, which contains virtually no useful information, but I'm pretty sure that you are hitting a BUG_ON(!list_empty(&req->r_req_lru_item)); in __kick_osd_requests(). In newer kernels this BUG_ON has been changed to a WARN_ON so it'll stamp out a warning instead of panicing, but the actual issue that causes this BUG_ON/WARN_ON to fire needs fixing - I'll bump #9779 (which is what's going to fix this) and start working on it.

I'd encourage you to not delete pools with mapped rbd images. It's OK to delete a pool to quickly remove unmapped images, but, like I said, doing this with mapped images while I/O from/to those images is in progress is just asking for trouble - at the very least you are leaving behind a useless block device with a bumped refcount and process(es) stuck in D state waiting for I/O. Use something like

```
while read DEV; do
    rbd unmap $DEV
done < <(rbd --format=json showmapped | python -c "
import sys, json;
for i in json.load(sys.stdin).itervalues():
    if i['pool'] == '$POOL':
        print i['device']
")
```

to unmap all mapped images from a given pool. That said, we won't panic and will print something like "pool foo existed but now does not" to dmesg after #9779 is resolved.

#7 - 06/22/2015 10:48 AM - Ilya Dryomov

A stripped down reproducer:

MON=1 OSD=1

```
$ cat notarget-pool-repro.sh
#!/bin/bash
rbd create --size 1 test
rbd map test
ceph osd pool delete rbd rbd --yes-i-really-really-mean-it
dd if=/dev/rbd0 of=/dev/null count=1 & # will block
pkill ceph-osd
```

#8 - 06/05/2016 06:24 PM - Ilya Dryomov

Fixed with #9779 in 4.7.

Files

after_removal.jpg	37.5 KB	06/16/2015	Alex Leake
panic.jpg	90.5 KB	06/16/2015	Alex Leake