rbd - Bug #2937

btrfs filesystem on rbd device kernel BUG writing large file

08/10/2012 01:59 PM - Bartek Kania

Status:	Duplicate	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Alex Elder		
Category:			
Target version:	v0.51		
Source:	Development	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:		**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

Writing a large file with dd on btrfs filesystem mounted from rbd device causes kernel bug

Stock kernel 3.5.1, config attached
ceph version 0.48argonaut-1~bpo60+1

Steps to reproduce:

```
rbd map testy --pool rbd --name client.admin --secret /etc/ceph/secret
mkfs.btrfs /dev/rbd/rbd/testy
mount /dev/rbd/rbd/testy /mnt
dd if=/dev/zero of=/mnt/1G bs=1M count=1024
```

Kernel BUG is attached.

History

#1 - 08/11/2012 06:31 PM - Sage Weil

- Priority changed from Normal to High

#2 - 08/12/2012 03:38 AM - Bartek Kania

I activated some extra debugging.
This appears just before the BUG:

```
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1449: rbd:  fetched request
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1472: rbd:  write 0x80000 bytes at 0x35720000
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1484: rbd:  rq->bio->bi_vcnt=128
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_do_request:890: rbd:  rbd_do_request obj=rb.0.1.0000000000d5 ofs=
524288 len=3276800
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1449: rbd:  fetched request
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1472: rbd:  write 0x80000 bytes at 0x357a0000
Aug 12 12:30:07 cephc kernel: [1486] rbd:rbd_rq_fn:1484: rbd:  rq->bio->bi_vcnt=128
Aug 12 12:30:07 cephc kernel: [1486] rbd:bio_chain_clone:731: rbd:  bio_chain_clone split! total=0 remaining=3
93216bi_size=524288
```

Hope this helps.

#3 - 08/16/2012 12:23 PM - Sage Weil

- Priority changed from High to Urgent

#4 - 08/16/2012 12:24 PM - Sage Weil

- Project changed from Ceph to rbd

- Category deleted (26)

#5 - 08/20/2012 02:09 PM - Sage Weil

- Target version set to v0.51

#6 - 08/20/2012 09:42 PM - Dan Mick

This reproduces on plana. Details: two machine cluster, one monitor, two OSDs:

```
roles:
- [mon.0, osd.0]
- [osd.1]

tasks:
- ceph:
- interactive:
```

4GB rbd image (old-format); no auth. Kernel ver 3.6.0-rc2-ceph-00140-g21b5c72

#7 - 08/28/2012 04:30 PM - Alex Elder

This smells a bit like it's related to this discussion:
https://patchwork.kernel.org/patch/1271871/

I terminated that discussion with a "you don't have it right but I haven't told you exactly how to fix it" message, so I take some responsibility for trying to get that fixed.

In any case, we don't use bio_split() the way anybody else does, and in fact as Guangliang Zhao pointed out, we leak a bio (or maybe a bio_pair, I don't remember). I'll have to take a little time refreshing my memory on what's going on there to be able to suggest how to fix it.

My suspicion is that fixing that problem might also lead to at least an explanation of how the but Dan is now able to reproduce is occurring.

#8 - 10/09/2012 09:47 PM - Alex Elder

```
That BUG_ON() call is this in the v3.5.1 kernel, in bio_split():
        BUG_ON(bi->bi_vcnt != 1);

The last few debug messages just prior to the crash indicated this:
    rbd:  write 0x80000 bytes at 0x357a0000
    rbd:rbd_rq_fn:1484: rbd:  rq->bio->bi_vcnt=128
    rbd:  bio_chain_clone split! total=0 remaining=393216bi_size=524288

So we got a 512KB write request at offset 0x357a0000 = sector 1752320.
The bio has 128 I/O vectors in it.

The first 128KB of the request have already been processed.  That means
the next offset is 1752320 + 256 sectors = 1752576 = 0x357c0000 bytes,
which is a reasonable split point.

I suspect that the split is occurring in a bio that has multiple
I/O vectors in it (as opposed to zero), which is not allowed.  I
think that's supposed to be handled by rbd_merge_bvec() (?) but
I'm honestly not sure what is supposed to guarantee that.
```

#9 - 10/09/2012 09:52 PM - Alex Elder

Also, given what I last wrote, I no longer think it's related
to the bio_chain_clone() problem mentioned earlier.

The latest fix for that (which is undergoing some testing right now)
might possibly avoid the problem of a multi-page bio getting through
to bio_split() (because it avoids bio_split()).

#10 - 10/10/2012 12:54 PM - Alex Elder

I looked at rbd_merge_bvec() this morning. I believe that,
given how it's used, it should be preventing callers from
adding a page to a bio if doing so would cross an rbd object
boundary.

It's not the best documented method, and this function is
not as clearly written as it could be. But basically it
should be returning how much data could be read/written
at a given point before hitting a "device" (or in our case,
an object) boundary. And the caller is supposed to **not**
add data to the bio if the result is less than the length
specified in the given bio_vec (the third argument).

#11 - 10/10/2012 03:22 PM - Dan Mick

- Assignee set to Alex Elder

#12 - 10/12/2012 09:18 AM - Alex Elder

Possibly related: http://tracker.newdream.net/issues/3291

#13 - 10/21/2012 08:50 AM - Sage Weil

- Status changed from New to 7

this was a btrfs bug; Josef has a fix pending.

#14 - 10/21/2012 08:52 AM - Sage Weil

- Status changed from 7 to Duplicate

Files

config.txt	62.6 KB	08/10/2012	Bartek Kania
bug.txt	3.28 KB	08/10/2012	Bartek Kania