bluestore - Bug #22464

Bluestore: many checksum errors, always 0x6706be76 (which matches a zero block)

12/17/2017 01:37 PM - Martin Preuss

Status:	Won't Fix	**Start date:**	12/17/2017
Priority:	Urgent	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	v12.2.1, v12.2.2
Backport:		**ceph-qa-suite:**	ceph-deploy
Regression:	No	**Pull request ID:**	
Severity:	2 - major		

Description

I'm new to Ceph. I started a ceph cluster from scratch on Debian 9,
consisting of 3 hosts, each host has 3-4 OSDs (using 4TB hdds, currently
totalling 10 hdds).

Right from the start I always received random scrub errors telling me
that some checksums didn't match the expected value, fixable with "ceph
pg repair".

I looked at the ceph-osd logfiles on each of the hosts and compared with
the corresponding syslogs. I never found any hardware error, so there
was no problem reading or writing a sector hardware-wise. Also there was
never any other suspicious syslog entry around the time of checksum
error reporting.

When I looked at the checksum error entries I found that the reported
bad checksum always was "0x6706be76".

Cluster created with version 12.2.1 (errors already existed with that version) and updated to 12.2.2.
All 3 nodes run Debian 9 with packages from "http://eu.ceph.com/debian-luminous/".

Cluster status:
services:
mon: 3 daemons, quorum ceph1,ceph2,ceph3
mgr: ceph1(active), standbys: ceph2
mds: cephfs-1/1/1 up {0=ceph1=up:active}, 2 up:standby
osd: 10 osds: 10 up, 10 in

```
data:
    pools:   5 pools, 256 pgs
    objects: 8097k objects, 10671 GB
    usage:   25403 GB used, 11856 GB / 37259 GB avail
    pgs:     256 active+clean
```

Pools:
pool 1 'cephfs_metadata' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 32 pgp_num 32 last_change 1184 flags hashpspool stripe_width 0 application cephfs
pool 2 'cephfs_data' replicated size 2 min_size 2 crush_rule 0 object_hash rjenkins pg_num 64 pgp_num 64 last_change 1184 lfor 0/772 flags hashpspool stripe_width 0 compression_algorithm zlib compression_mode force application cephfs
pool 3 'cephfs_home' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 32 pgp_num 32 last_change 1184 lfor 0/463 flags hashpspool stripe_width 0 compression_algorithm zlib compression_mode force application cephfs
pool 4 'cephfs_multimedia' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 64 pgp_num 64 last_change 1184 lfor 0/705 flags hashpspool stripe_width 0 application cephfs

pool 5 'cephfs_vdr' replicated size 2 min_size 1 crush_rule 0 object_hash rjenkins pg_num 64 pgp_num 64 last_change 1184 lfor 0/632 flags hashpspool stripe_width 0 application cephfs

OSD tree:

```
ID CLASS WEIGHT   TYPE NAME      STATUS REWEIGHT PRI-AFF
-1       36.38596 root default
-3       10.91579     host ceph1
0   hdd  3.63860         osd.0      up  0.79999 1.00000
1   hdd  3.63860         osd.1      up  0.70000 1.00000
2   hdd  3.63860         osd.2      up  1.00000 1.00000
-5       14.55438     host ceph2
3   hdd  3.63860         osd.3      up  1.00000 1.00000
4   hdd  3.63860         osd.4      up  1.00000 1.00000
5   hdd  3.63860         osd.5      up  1.00000 1.00000
9   hdd  3.63860         osd.9      up  1.00000 1.00000
-7       10.91579     host ceph3
6   hdd  3.63860         osd.6      up  1.00000 1.00000
7   hdd  3.63860         osd.7      up  1.00000 1.00000
8   hdd  3.63860         osd.8      up  1.00000 1.00000
```

Related issues:

Related to bluestore - Bug #22102: BlueStore crashed on rocksdb checksum mism...	**Won't Fix**	**11/10/2017**
Related to bluestore - Bug #25006: bad csum during upgrade test	**Can't reproduce**	**07/19/2018**

History

#1 - 12/17/2017 01:40 PM - Martin Preuss

Excerpt from ceph osd logfiles:

2017-12-10 02:48:43.948386 7fed88c8a700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xa2fc307f, device location [0x2f7de040000~1000], logical extent 0x0~1000, object #4:6ed0f2be:::100000086c5.000000ab:head#
2017-12-10 02:56:45.417924 7fed88c8a700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x91d3e073, device location [0x508c720000~1000], logical extent 0x0~1000, object #5:c826bc6a:::100002cbbc1.000000a0:head#
2017-12-08 03:01:04.497951 7fed8a48d700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x6fc5414c, device location [0x21f871b0000~1000], logical extent 0x280000~1000, object #5:27c2eefc:::10000009e03.000002c6:head#
2017-12-08 03:05:17.892672 7fed88c8a700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x40000, got 0x6706be76, expected 0x3e982845, device location [0xc939ed0000~1000], logical extent 0x40000~1000, object #4:70b8d408:::10000009076.000000d8:head#
2017-12-06 02:51:18.307194 7fed8948b700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x9afdde9, device location [0x125752a0000~1000], logical extent 0x300000~1000, object #4:0c825688:::1000000909f.0000008e:head#
2017-12-03 11:06:09.185188 7fd7d16c2700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x40f82988, device location [0x161d6400000~1000], logical extent 0x0~1000, object #5:4135e934:::10000009e45.000001e1:head#
2017-12-03 11:20:18.664675 7fd7d16c2700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x60ef9e8d, device location [0x4c25a70000~1000], logical extent 0x0~1000, object #5:432fbeae:::100002acc32.000001e6:head#
2017-12-03 11:31:55.395281 7fd7d26c4700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xacf29cfe, device location [0x6366850000~1000], logical extent 0x0~1000, object #5:b4d3b8c1:::100002cc114.00000082:head#
2017-12-03 11:54:47.385602 7fd7d26c4700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xff725cb6, device location [0x35cdb4c0000~1000], logical extent 0x300000~1000, object #5:b7b5c9fd:::100002ad208.000001d4:head#

2017-12-10 01:21:07.506122 7f17fd870700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x192e1d28, device location [0x14095ed0000~1000], logical extent 0x200000~1000, object #4:ce07cedb:::10000008641.00000142:head#
2017-12-10 02:06:06.682700 7f17fc86e700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x41a2bc4c, device location [0x348c6380000~1000], logical extent 0x200000~1000, object #3:12a853c6:::100001cfa81.000004c8:head#
2017-12-07 02:07:27.693073 7f17fd870700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x5f0bce3f, device location [0x2e06f300000~1000], logical extent 0x380000~1000, object #3:5b96f1f3:::1000021f0aa.00000297:head#

2017-12-09 01:42:47.915186 7f0fc370e700 -1 bluestore(/var/lib/ceph/osd/ceph-2) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x330b1279, device location [0x2b8aa440000~1000], logical extent 0x380000~1000, object #5:5168ad49:::100002acb4a.000001c4:head#
2017-12-03 11:01:17.808106 7f78eba01700 -1 bluestore(/var/lib/ceph/osd/ceph-2) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xfcb0906, device location [0x2fed9920000~1000], logical extent 0x100000~1000, object

#4:8a73381f:::1000000822f.00000028:head#
2017-12-03 11:12:47.971419 7f78eba01700 -1 bluestore(/var/lib/ceph/osd/ceph-2) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xe3fb194b, device location [0x10a22510000~1000], logical extent 0x200000~1000, object #5:6933f50f:::1000000a2f1.00000247:head#
2017-12-03 11:31:55.363014 7f78eba01700 -1 bluestore(/var/lib/ceph/osd/ceph-2) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xe2770519, device location [0x6d76b40000~1000], logical extent 0x180000~1000, object #5:6bcfca44:::10000009747.00000017:head#

#2 - 01/18/2018 03:30 PM - Sage Weil

- Subject changed from Bluestore - random checksum errors to Bluestore - many checksum errors, always 0x6706be76

- Priority changed from Normal to High

#3 - 01/19/2018 07:12 PM - Martin Preuss

- File ceph-errors added

Just an update:

"ceph pg repair x.yz"

changes the ceph status from HEALTH_ERR to HEALTH_OK (I have to do that everyday now), but at the next deep scrub of the same pg the same checksum error appears on some of the pgs, so "ceph pg repair" doesn't really seem to repair anything, at least with some pgs...

Everyday for a few weeks now I add the list of inconsistent pgs (output of "ceph health detail") to a file, separated by a blank line for every day.
I appended that file to the ticket. As you can see some pgs are mentioned multiple times.

#4 - 01/22/2018 03:34 PM - Sage Weil

Martin, can you check your dmesg/kernel log and see if there are any media errors? The crc value is for a single block of all zeros. We're not sure why the device would be returning that...

#5 - 01/23/2018 12:23 PM - Adam Kupczyk

Martin, your logs show places where data is located, for example: "device location [0x6d76b40000~1000]".
Is it possible to read contents of your disk in those places to verify if this locations contain only zeros?

#6 - 01/23/2018 10:24 PM - Martin Preuss

Hi,

how do I translate the given location, e.g. to a "dd" argument?

Meanwhile I found out that only the first machine - ceph1 - shows the crc32 error, and all problematic pgs show involvement of one of the OSDs hosted on ceph1. Logs on other hosts (ceph2- ceph3) don't show a crc error, they only state that there was a read error like in

5.2 shard 0: soid 5:406c1cf7:::100002d4272.00001781:head candidate had a read error

So the real problem seems to only be on ceph1... However, I still don't see any suspicious log entries in syslog, there is no media error reported...

Regards
Martin

#7 - 01/25/2018 02:10 PM - Adam Kupczyk

Martin,

For "device location [0x6d76b40000~1000]" it would be:
dd bs=4096 if=/var/lib/ceph/osd/ceph-1/block skip=$(printf %d 0x6d76b40) count=1 of=data.bin
or similar, depending where your block device with data is located.

Best regards,
Adam

#8 - 01/31/2018 08:23 AM - Nicolas Drufin

I have the same problem on my cluster. Periodically I got pg inconsistent only on bluestore osd with this type of message :

```
2018-01-30 23:30:56.514373 7f7490d15700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000
checksum at blob offset 0x30000, got 0x6706be76, expected 0xcb511292, device location [0x3b874a0000~1000], log
ical extent 0x230000~1000, object #6:3b6fafa2:::rbd_data.4508774b0dc51.0000000000000515:head#
```

When I repair it, it disapears.

I have made dd command to test after repair and the result seems to be some string. I will try before cepg pg repair next time.

#9 - 02/02/2018 03:08 AM - Paul Emmerich

I'm also seeing this on one cluster. Bluestore and CephFS, replicated pools, no compression, HDDs.
It happens randomly across 10 different servers on different disks, several PGs per day where each PG reports between 1 and 3 incorrect checksums claiming that the checksum is 0x6706be76.

I manually dumped the data from the disk before at the position indicated in the log file and it's some data, i.e. not all zeroes as the checksum suggests.
Running repair reports "repair ok, 0 fixed" and the data on disk is not modified.

The cluster seems to work fine otherwise.

#10 - 02/02/2018 01:29 PM - Paul Emmerich

It doesn't seem to happen on all servers, it's only 5 out of 15.
But there is nothing special about the affected servers, there are others with identical hardware and software that aren't affected.

Hardware is HP with a smartarray als jbod, software is Ceph 12.2.2 (cf0baeeeeba3b47f9427c6c97e2144b094b7e5ba) on Debian Stretch with Kernel 4.14.

#11 - 02/03/2018 01:08 AM - Martin Preuss

I just re-created all 3 OSDs on ceph1 (the host which had the read errors).

Now the errors occur less often, but they still persist.

I finally got the opportunity to use the command proposed by Adam a few posts ago.

ceph health detail shows this:

```
    pg 4.2d is active+clean+inconsistent, acting [2,6,9]
```

The logfile /var/log/ceph/ceph-osd.2.log shows this:

```
2018-02-03 01:00:00.486305 7f7b13e0b700  0 log_channel(cluster) log [DBG] : 4.2d deep-scrub starts
2018-02-03 01:00:30.928758 7f7b13e0b700 -1 bluestore(/var/lib/ceph/osd/ceph-2) _verify_csum bad crc32c/0x1000
checksum at blob offset 0x0, got 0x6706be76, expected 0xdd2a27c2, device location [0x15438e0000~1000], logical
 extent 0x0~1000, object #4:b43d74bb:::10000008608.00000088:hea
2018-02-03 01:00:31.089325 7f7b13e0b700 -1 log_channel(cluster) log [ERR] : 4.2d shard 2: soid 4:b43d74bb:::10
000008608.00000088:head candidate had a read error
2018-02-03 01:08:17.658957 7f7b13e0b700 -1 log_channel(cluster) log [ERR] : 4.2d deep-scrub 0 missing, 1 incon
sistent objects
2018-02-03 01:08:17.658980 7f7b13e0b700 -1 log_channel(cluster) log [ERR] : 4.2d deep-scrub 1 errors
```

So I gather the location is 0x15438e0000. So the command I used to read that particular block was this:

```
dd bs=4096 if=/var/lib/ceph/osd/ceph-2/block skip=$(printf %d 0x15438e0) count=1 of=/tmp/after-data.bin
```

I did this before "ceph pg repair" and after. In both cases the data is the same. Also, the data block contains random data, not all zeroes.

And of course there still is no hint in the syslog about any hardware error on the disk used by osd.2 (or any hardware problem log) ...

BTW: This is the output of rados list-inconsistent-obj before repair:

```
#> rados list-inconsistent-obj 4.2d --format=json-pretty
{
    "epoch": 2731,
    "inconsistents": [
        {
            "object": {
                "name": "10000008608.00000088",
                "nspace": "",
                "locator": "",
                "snap": "head",
                "version": 24561
            },
            "errors": [],
            "union_shard_errors": [
                "read_error"
            ],
            "selected_object_info": "4:b43d74bb:::10000008608.00000088:head(72'24561 client.24252.1:367516 dir
ty|data_digest|omap_digest s 4194304 uv 24561 dd deba5ce4 od ffffffff alloc_hint [0 0 0])",
            "shards": [
                {
                    "osd": 2,
                    "primary": true,
                    "errors": [
                        "read_error"
                    ],
                    "size": 4194304
                },
                {
                    "osd": 6,
                    "primary": false,
                    "errors": [],
                    "size": 4194304,
```

```
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xdeba5ce4"
                },
                {
                    "osd": 9,
                    "primary": false,
                    "errors": [],
                    "size": 4194304,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xdeba5ce4"
                }
            ]
        }
    ]
}
```


#12 - 02/06/2018 10:42 PM - Martin Preuss

Update: Now at least one other host starts giving me these crc errors, too...

So I have now at least two out of three hosts with these crc errors. All hosts are equipped with the same hardware. I guess if I wait long enough I'll get the last one to error out as well :-(...

#13 - 02/11/2018 05:27 PM - Paul Emmerich

Martin, let's compare hardware. The only cluster I'm seeing this on is HP servers with smartarray controllers and lots of 2.5" disks.

#14 - 02/16/2018 05:24 PM - Martin Preuss

Hi,

now I see an inconsistent PG for which both OSDs report that HEAD has a read error, but in this case no read error is logged on any machine...

PG_DAMAGED Possible data damage: 1 pg inconsistent
pg 5.60 is active+clean+inconsistent, acting [9,7]

osd.9:

2018-02-16 02:55:04.398415 osd.9 osd.9 192.168.115.142:6803/31800 1500 : cluster [ERR] 5.60 shard 7: soid 5:07ff1ee6:::1000036a3ff.0000017d:head candidate had a read error
2018-02-16 02:55:15.319626 mon.ceph1 mon.0 192.168.115.141:6789/0 14955 : cluster [ERR] Health check failed: 1 scrub errors (OSD_SCRUB_ERRORS)
2018-02-16 02:55:15.319695 mon.ceph1 mon.0 192.168.115.141:6789/0 14956 : cluster [ERR] Health check failed: Possible data damage: 1 pg inconsistent (PG_DAMAGED)
2018-02-16 02:55:12.533078 osd.9 osd.9 192.168.115.142:6803/31800 1501 : cluster [ERR] 5.60 deep-scrub 0 missing, 1 inconsistent objects
2018-02-16 02:55:12.533087 osd.9 osd.9 192.168.115.142:6803/31800 1502 : cluster [ERR] 5.60 deep-scrub 1 errors
2018-02-16 02:56:01.512627 mon.ceph1 mon.0 192.168.115.141:6789/0 14957 : cluster [ERR] overall HEALTH_ERR 1 scrub errors; Possible data damage: 1 pg inconsistent

osd.7:

2018-02-16 02:55:04.398408 7f3c55b04700 -1 log_channel(cluster) log [ERR] : 5.60 shard 7: soid 5:07ff1ee6:::1000036a3ff.0000017d:head candidate had a read error
2018-02-16 02:55:12.533074 7f3c55b04700 -1 log_channel(cluster) log [ERR] : 5.60 deep-scrub 0 missing, 1 inconsistent objects
2018-02-16 02:55:12.533085 7f3c55b04700 -1 log_channel(cluster) log [ERR] : 5.60 deep-scrub 1 errors

I would like to try with filestore, but since the introduction of this ceph-volume stuff creating an osd has become much more complicated compared to just using
ceph-deploy osd prepare cephX:sdY (why is that??!).

However, I can't figure out how to specify the desired storage type with ceph-deploy, it always uses bluestore. So I'm stuck with bluestore for now... :-(

#15 - 02/16/2018 05:32 PM - Martin Preuss

Hi Paul,

Paul Emmerich wrote:

> Martin, let's compare hardware. The only cluster I'm seeing this on is HP servers with smartarray controllers and lots of 2.5" disks.

I have a small cluster (for evaluation purposes, but in active use) consisting of just 3 machines, all have more or less the same hardware:
- ASRockRack C236 WSI
- Intel Core i3-6100
- 8-16GB of RAM
- 3-4 hdds per host, each with 4TB capacity
- operating system on an extra SSD

#16 - 02/19/2018 05:47 PM - Sage Weil

Martin Preuss wrote:

> I would like to try with filestore, but since the introduction of this ceph-volume stuff creating an osd has become much more complicated compared to just using
> ceph-deploy osd prepare cephX:sdY (why is that??!).
>
> However, I can't figure out how to specify the desired storage type with ceph-deploy, it always uses bluestore. So I'm stuck with bluestore for now... :-(

I think you just need to add --filestore to the command?

#17 - 02/19/2018 05:47 PM - Sage Weil

- Subject changed from Bluestore - many checksum errors, always 0x6706be76 to Bluestore: many checksum errors, always 0x6706be76 (which matches a zero block)

- Status changed from New to Verified

#18 - 02/22/2018 11:15 PM - Adam Kupczyk

Hi Martin,

I am not sure yet what causes problem with 0x6706be76 crc.
To pinpoint, I added debug code to close in on problem:
https://github.com/ceph/ceph-ci/tree/wip-22464-reread-check ,
which is based on 12.2.2.

You can fetch and compile, or try OSD from build packages:
https://shaman.ceph.com/repos/ceph/wip-22464-reread-check/1c11b2cb3aa62a6c8df695f697f67fa8d81af9ba/

After "_verify_csum bad" there will be 3 additional actions noted in logs:
1) dump content of already read data
2) dump of data read from disk again
3) retry of data read

It may happen, that due to 3) this will hide any problem.
This change will never go to production, so if your log shows "_verify_csum bad" please notify, will continue solving.

Best regards,
Adam Kupczyk

#19 - 02/27/2018 08:50 PM - Martin Preuss

I recently upgraded one of my test nodes, now two of my three node have 16 gb RAM with 4 OSDs (4tb hdd each), the 3rd node has still only 8gb but only 3 OSDs (4tb hdd each).

Since those two hosts no longer use swap space the error didn't yet occur for some days now, before that I had at least one error every day...

#20 - 03/03/2018 02:55 PM - Paul Emmerich

Tl;dr: retrying the read works.

I've also been running one server with that patch for a few days and got a log for you.

The OSDs on that server encountered 48 scrub errors (out of a total of 200 over the whole cluster).

27 of the scrub errors are "just"

```
Mar 03 04:51:42 XXXX ceph-osd[9974]: 2018-03-03 05:51:42.451580 7f80d347b700 -1 log_channel(cluster) log [ERR]
 : 2.79 shard 61: soid 2:9e07e948:::10000131559.000001e1:head candidate had a read error
```

and don't trigger the patch, but I guess the root cause here is the same (disks are okay and were running fine with filestore).

But one triggered the 0x6706be76 checksum:

```
Mar 03 02:33:25 XXX ceph-osd[20317]: 2018-03-03 03:33:25.472804 7f7ce72aa700 -1 bluestore(/var/lib/ceph/osd/ceph-48) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x1000, got 0x6706be76, expected 0x3b43b141, device location [0x689e681000~1000], logical extent 0x381000~1000, object #2:81c1b635:::100001389a3.00005ef1:head#
Mar 03 02:33:25 XXX ceph-osd[20317]: 2018-03-03 03:33:25.472917 7f7ce72aa700 -1 bluestore(/var/lib/ceph/osd/ceph-48) _do_read failed checksum, retrying read
Mar 03 02:33:25 XXX ceph-osd[20317]: 2018-03-03 03:33:25.475154 7f7ce72aa700 -1 bluestore(/var/lib/ceph/osd/ceph-48) _do_read retried read succeeds
```

Is retrying the read once a good idea in general if the checksum doesn't match?

Regarding Martin's comment about swap: we don't do swap on our servers (PXE booted and OS running from RAM), so that's not the root cause.

#21 - 03/04/2018 04:15 PM - Paul Emmerich

Used the wrong OSDs for the previous post (these servers mostly had the "head candidate had a read error" scrub error), I've now tested another server.

I've now got 350 log entries with checksum 0x6706be76 and the retried read succeeded with 349 of them.
There is no obvious pattern in the location or offset in the object.

Here's the log were the retry didn't succeed.

```
2018-03-04 08:53:15.444094 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x7000, got 0x6706be76, expected 0xe9f9afad, device location [0x1be45b87000~1000], logical extent 0x87000~1000, object #2:e76fb545:::10000000c20.0000025c:head#
2018-03-04 08:53:15.444117 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) _do_read failed checksum, retrying read
2018-03-04 08:53:15.458431 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xa515a52c, device location [0x1be45b80000~1000], logical extent 0x80000~1000, object #2:e76fb545:::10000000c20.0000025c:head#
2018-03-04 08:53:15.458441 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) _do_read retried read failed, giving up
2018-03-04 08:53:15.459842 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) Data from _do_read:
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000ff0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00001000

2018-03-04 08:53:15.460641 7f27dcdae700 -1 bluestore(/var/lib/ceph/osd/ceph-112) Data from device [0x1be45b80000~1000] res=0
00000000  XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX XX  |...<..|/Z.l...:.|
(data redacted, but it looks reasonable and is definitely not all 0)
00001000
```

I've previously said that this happens on about 5 out of 15 servers, this was just a guesstimate.

I've now looked into the logs of two nights of scrubbing to maybe find some pattern.

We've got 12 servers with HDDs here, number of disks per server varies between 10 and 20.
8 of the servers have identical hardware and software (Debian 9, exact same image booted via PXE).
I've even checked the firmware versions of the smart array controllers; there are two different versions (not tied to failing servers).
All caches are disabled.

3 of these 8 servers are affected and all of the disks report errors. Two of them mostly reports lots of "head candidate had a read error", one lots of checksum 0x6706be76 errors.
All of them worked fine with Filestore, the hardware incl. disks were previously used in ZFS raids which were scrubbed regularly with no problems.

#22 - 03/05/2018 03:51 PM - Marco Baldini

I think I'm having similar problem in my 3 nodes ceph cluster.
It's installed on proxmox nodes, each with 3x1TB HDD and 1x240GB SSD. I created this cluster after Luminous release, so all OSDs are Bluestore.

I randomly get OSD_SCRUB_ERRORS after scrub (I set it during the night and I check ceph health in the morning) with 1 or 2 random PGs, in different OSDs and from different nodes.

Checking in the OSD logs for the most recent PGs with errors, I see:

OSD.5 (in host 2)

```
2018-03-01 20:25:46.804384 7fdf4d515700  0 log_channel(cluster) log [DBG] : 9.1c deep-scrub starts
- cut -
2018-03-01 20:30:02.702269 7fdf4d515700  2 osd.5 pg_epoch: 16486 pg[9.1c( v 16486'51798 (16431'50251,16486'517
98] local-lis/les=16474/16475 n=3629 ec=1477/1477 lis/c 16474/16474 les/c/f 16475/16477/0 16474/16474/16474) [
5,6] r=0 lpr=16474 crt=16486'51798 lcod 16486'51797 mlcod 16486'51797 active+clean+scrubbing+deep] 9.1c shard
6: soid 9:3b157c56:::rbd_data.1526386b8b4567.0000000000001761:head candidate had a read error
2018-03-01 20:30:02.702278 7fdf4d515700 -1 log_channel(cluster) log [ERR] : 9.1c shard 6: soid 9:3b157c56:::rb
d_data.1526386b8b4567.0000000000001761:head candidate had a read error
- cut -
2018-03-01 20:31:21.536849 7fdf4d515700 -1 log_channel(cluster) log [ERR] : 9.1c deep-scrub 0 missing, 1 incon
sistent objects
2018-03-01 20:31:21.536852 7fdf4d515700 -1 log_channel(cluster) log [ERR] : 9.1c deep-scrub 1 errors
```

OSD.4 (in host 2)

```
2018-02-28 00:03:06.157194 7f112cf76700  0 log_channel(cluster) log [DBG] : 13.65 deep-scrub starts
2018-02-28 00:03:33.458558 7f112cf76700 -1 log_channel(cluster) log [ERR] : 13.65 shard 2: soid 13:a719ecdf:::
rbd_data.5f65056b8b4567.000000000000f8eb:head candidate had a read error
2018-02-28 00:03:59.061512 7f112cf76700 -1 log_channel(cluster) log [ERR] : 13.65 deep-scrub 0 missing, 1 inco
nsistent objects
2018-02-28 00:03:59.061519 7f112cf76700 -1 log_channel(cluster) log [ERR] : 13.65 deep-scrub 1 errors
```

OSD.8 (in host 3)

```
2018-02-27 23:55:01.695849 7f4dd0816700  0 log_channel(cluster) log [DBG] : 14.31 deep-scrub starts
2018-02-27 23:55:15.100084 7f4dd0816700 -1 log_channel(cluster) log [ERR] : 14.31 shard 1: soid 14:8cc6cd37:::
rbd_data.30b15b6b8b4567.00000000000081a1:head candidate had a read error
2018-02-27 23:55:29.725097 7f4dd0816700 -1 log_channel(cluster) log [ERR] : 14.31 deep-scrub 0 missing, 1 inco
nsistent objects
2018-02-27 23:55:29.725107 7f4dd0816700 -1 log_channel(cluster) log [ERR] : 14.31 deep-scrub 1 errors
```

After a ceph pg repair, health is back OK, in OSD log I see <PG> repair starts then <PG> repair ok, 0 fixed

My ceph versions


```
{
    "mon": {
        "ceph version 12.2.2 (215dd7151453fae88e6f968c975b6ce309d42dcf) luminous (stable)": 3
    },
    "mgr": {
        "ceph version 12.2.2 (215dd7151453fae88e6f968c975b6ce309d42dcf) luminous (stable)": 3
    },
    "osd": {
        "ceph version 12.2.2 (215dd7151453fae88e6f968c975b6ce309d42dcf) luminous (stable)": 12
    },
    "mds": {},
    "overall": {
        "ceph version 12.2.2 (215dd7151453fae88e6f968c975b6ce309d42dcf) luminous (stable)": 18
    }
}
```


#23 - 03/06/2018 09:30 AM - Adam Kupczyk

Hi Paul,

Currently, I am continuing research on "crc 0x6706be76" issue. Its relations to deep-scrub errors will be handled separately.

I posted change to bluestore operation. New version drops using AIO for reads entirely.
Logic for re-reading is still there. Please check if this suppresses the problem.
https://shaman.ceph.com/builds/ceph/wip-22464-reread-check/

Best regards,
Adam

#24 - 03/07/2018 02:39 PM - Paul Emmerich

We didn't run any deep scrubs the last day or so due to a longer backfill, will report back later.

#25 - 03/13/2018 04:27 PM - Eric Blevins

I am seeing this issue too.
We were running Proxmox 4.x with CEPH 12.2.2 until a few weeks ago, never had a problem.
Upgraded to 12.2.3 a few weeks ago, still had no issues.
Last week we upgraded to 12.2.4 and also upgraded to Proxmox to 5.x

Seen the issue four times since the 9th:

```
Host A:
2018-03-09 05:58:12.273827 7f51e9024700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad crc32c/0x1000
checksum at blob offset 0x0, got 0x6706be76, expected 0x7ad05e25, device location [0x3280ab60000~1000], logica
l extent 0x0~1000, object #2:dc60ca55:::rbd_data.197267b2ae8944a.0000000000036210:head#
2018-03-13 10:34:01.271346 7f2ffddec700 -1 bluestore(/var/lib/ceph/osd/ceph-11) _verify_csum bad crc32c/0x1000
 checksum at blob offset 0x2d000, got 0x6706be76, expected 0x92534989, device location [0x3192048d000~1000], l
ogical extent 0x12d000~1000, object #2:d83b5270:::rbd_data.1ba62b22ae8944a.0000000000033c2e:head#

Host B:
2018-03-12 05:16:53.472938 7f83a5990700 -1 bluestore(/var/lib/ceph/osd/ceph-10) _verify_csum bad crc32c/0x1000
 checksum at blob offset 0x0, got 0x6706be76, expected 0xe6979aae, device location [0x28124f00000~1000], logic
al extent 0x80000~1000, object #2:4a16beca:::rbd_data.1ba62b22ae8944a.00000000000faedb:head#

Host C:
2018-03-12 05:45:50.847982 7f71e7f51700 -1 bluestore(/var/lib/ceph/osd/ceph-18) _verify_csum bad crc32c/0x1000
 checksum at blob offset 0x0, got 0x6706be76, expected 0x71b2d931, device location [0x5d73d90000~1000], logica
l extent 0x180000~1000, object #2:f77a13c6:::rbd_data.b84ef56b8b4567.0000000000000358:head#
```

When upgrading Proxmox I moved from a 4.4 kernel to a 4.13 kernel.
Is it possible the source of this problem is in the kernel?

#26 - 03/14/2018 08:29 AM - Adam Kupczyk

Hi Eric,
I am trying to pinpoint whether problem is related to AIO. Have you tested on official ceph builds, or tried on those from
https://shaman.ceph.com/builds/ceph/wip-22464-reread-check/ ?
Adam

#27 - 03/20/2018 10:08 PM - Brian Marcotte

I'm seeing the same problem here.

When I get the notification about the deep scrub error, I don't need to do "repair", I just tell Ceph to do another deep scrub on the PG. The error clears a few minutes later.

The kernel does not report any I/O errors. Ceph reports the same errors others here reported:

```
... _verify_csum bad crc32c/0x1000 checksum ... got 0x6706be76, expected 0x6a702cfa ...
```

```
   ... cluster [ERR] ... head candidate had a read error
```

I was running 12.2.2 and upgraded to 12.2.4 which didn't fix the problem.

The cluster is made of three machines all running Debian 9 with the Debian kernel (4.9). Only the one machine which has bluestore OSDs is seeing this problem.

Thanks.

--
- Brian

#28 - 04/11/2018 12:51 PM - Michael Prokop

We're seeing the same behavior:

```
# zgrep 6706be76 ~log/ceph/ceph*
/var/log/ceph/ceph-osd.24.log.5.gz:2018-04-07 03:43:08.255385 7ffb50837700 -1 bluestore(/var/lib/ceph/osd/ceph
-24) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x1000, got 0x6706be76, expected 0x237ce1a, device
 location [0x1977e761000~1000], logical extent 0x1000~1000, object #8:0e6a4e66:::rbd_data.30d2182ae8944a.00000
000000423e3:head#
/var/log/ceph/ceph-osd.24.log.5.gz:2018-04-07 04:44:35.139173 7ffb5283b700 -1 bluestore(/var/lib/ceph/osd/ceph
-24) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x7e000, got 0x6706be76, expected 0xd42cfe28, devi
ce location [0x1d1a72fe000~1000], logical extent 0xfe000~1000, object #8:b9cf673b:::rbd_data.33152d74b0dc51.00
000000000340e1:head#
/var/log/ceph/ceph-osd.25.log.7.gz:2018-04-05 04:42:12.474493 7fa546fd5700 -1 bluestore(/var/lib/ceph/osd/ceph
-25) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x1000, got 0x6706be76, expected 0xe3ba3609, devic
e location [0x15fa5c11000~1000], logical extent 0x1000~1000, object #8:4b4546a6:::rbd_data.30d2182ae8944a.0000
000000043455:head#
/var/log/ceph/ceph-osd.26.log.1.gz:2018-04-11 02:15:09.104466 7f0715d53700 -1 bluestore(/var/lib/ceph/osd/ceph
-26) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x5c000, got 0x6706be76, expected 0xddbd2bd8, devi
ce location [0x149fadec000~1000], logical extent 0x25c000~1000, object #8:b2ae230d:::rbd_data.30d2182ae8944a.0
0000000000052b5:head#
```

We're using ceph 12.2.4 (upgrade history: 10.2.7->10.2.9->12.2.0->12.2.1->12.2.4), and have a mixture of filestore and bluestore on our 27 OSDs (distributed on three nodes) using 1024 PGs.

The bluestore OSDs were added with ceph v12.2.1. The first occurrence of this checksum/PG inconsistent issue appeared with ceph v12.2.1, though we're still seeing the same issue with v12.2.4.

OSDs 0-17 are running on filestore, OSDs 18-26 are running on bluestore. We can't identify any problems related to the underlying disks (SMART doesn't report any problems).

```
# ceph health detail
HEALTH_ERR 2 scrub errors; Possible data damage: 2 pgs inconsistent
OSD_SCRUB_ERRORS 2 scrub errors
PG_DAMAGED Possible data damage: 2 pgs inconsistent
    pg 8.14d is active+clean+inconsistent, acting [21,26,2]
    pg 8.204 is active+clean+inconsistent, acting [25,20,6]

# rados list-inconsistent-obj 8.204 --format=json-pretty
{
    "epoch": 2235,
    "inconsistents": [
        {
            "object": {
                "name": "rbd_data.eb255a2ae8944a.0000000000000cf4",
                "nspace": "",
```

```
                "locator": "",
                "snap": "head",
                "version": 3100116
            },
            "errors": [],
            "union_shard_errors": [
                "read_error"
            ],
            "selected_object_info": "8:20710bb9:::rbd_data.eb255a2ae8944a.0000000000000cf4:head(3760'3100116 c
lient.15414363.0:742606 dirty|data_digest|omap_digest s 4194304 uv 3100116 dd 1314d827 od ffffffff alloc_hint
[4194304 4194304 0])",
            "shards": [
                {
                    "osd": 6,
                    "primary": false,
                    "errors": [],
                    "size": 4194304,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0x1314d827"
                },
                {
                    "osd": 20,
                    "primary": false,
                    "errors": [
                        "read_error"
                    ],
                    "size": 4194304
                },
                {
                    "osd": 25,
                    "primary": true,
                    "errors": [],
                    "size": 4194304,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0x1314d827"
                }
            ]
        }
    ]
}
```


Any hints about what we could do as a fix/workaround?
Can we provide any further details that would help investigating this issue?
Is it known what might cause this issue?

#29 - 04/11/2018 02:36 PM - Eric Blevins

Could this be caused by re-written data while the first write is still in flight?

This write pattern has been observed as causing checksum errors in DRBD:
https://lists.gt.net/drbd/users/21069#21069

#30 - 04/17/2018 12:30 PM - Björn Lässig

Our Ceph Cluster has the same Problems. I am just migrating OSDs from filestore with XFS to bluestore and get these Errors

```
ceph-osd.0.log:2018-04-17 06:38:32.693980 7f70cd416700 -1 bluestore(/var/lib/ceph/osd/ceph-0) _verify_csum bad
 crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xe60c15fe, device location [0x2f52ef3000
0~1000], logical extent 0x180000~1000, object #3:fdb1a223:::rb.0.51010f.238e1f29.0000000180fe:head#
ceph-osd.0.log:2018-04-17 06:41:53.186254 7f70d6428700  4 rocksdb: [/build/ceph-12.2.4/src/rocksdb/db/db_impl_
write.cc:684] reusing log 9216 from recycle list
```

My cluster is running on Debian Stretch Backports Kernel 4.14.13 with ceph Version ceph version 12.2.4 (52085d5249a80c5f5121a76d6288429f35e4e77b) luminous (stable).

What can i do to help?

#31 - 04/18/2018 08:20 AM - Christoph Glaubitz

We also see the problem on two clusters with linux 4.13, but not on a cluster with linux 4.10. Configuration and ceph version is the same everywhere.

ceph-12.2.4

#32 - 04/24/2018 08:20 AM - Christoph Glaubitz

Just a follow up: We downgraded the kernel end of last week, and didn't get the scrub error any more. While before, it was around 3 times a day.

Now running

```
Linux DE-ES-001-03-09-12 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 x
86_64 GNU/Linux
```

kernel with with we saw scrub issues

```
Linux DE-ES-001-04-10-18 4.13.0-36-generic #40~16.04.1-Ubuntu SMP Fri Feb 16 23:25:58 UTC 2018 x86_64 x86_64 x
86_64 GNU/Linux
```

We still run 4.13 in a lab environment. So if we can support to tackle this down in any ways, please let me know.

#33 - 04/24/2018 03:26 PM - Sage Weil

This bug is starting to sound like #22102, which **looks** like pread() is getting zeros.

Cristoph, does the machine in question have swap enabled?

#34 - 04/24/2018 03:26 PM - Sage Weil

- Related to Bug #22102: BlueStore crashed on rocksdb checksum mismatch added

#35 - 04/24/2018 03:26 PM - Sage Weil

- Priority changed from High to Urgent

#36 - 04/24/2018 03:52 PM - Marco Baldini

I'm having the same problem on a 3 nodes ceph cluster, all three nodes have swap enabled and used, not too much, but even with swappiness = 0 swap is used a bit even with free memory.
I just issued swapoff -a on all the three nodes and I'll check if there will still be scrub errors

#37 - 04/24/2018 06:08 PM - Paul Emmerich

Update from the cluster where I saw this: the problem suddenly disappeared after a few seemingly random changes to the cluster (added/removed disks, rebooted the affected hosts).
I haven't seen an error since my last post where I mentioned that it was currently rebalancing/recovering and scrubbing was disabled.

We do not have swap enabled there (OS is running from RAM).

#38 - 04/26/2018 06:56 PM - Sage Weil

Paul Emmerich wrote:

> Update from the cluster where I saw this: the problem suddenly disappeared after a few seemingly random changes to the cluster (added/removed disks, rebooted the affected hosts).
> I haven't seen an error since my last post where I mentioned that it was currently rebalancing/recovering and scrubbing was disabled.
>
> We do not have swap enabled there (OS is running from RAM).

Paul: was swap ever enabled on the node(s) where you saw the issue?

#39 - 04/26/2018 08:54 PM - Paul Emmerich

No, never. We initially deployed the OSDs with Filestore on kernel 4.9 and then switched to Bluestore and kernel 4.14.

#40 - 04/27/2018 07:18 AM - Michael Prokop

Sage Weil wrote:

> Paul: was swap ever enabled on the node(s) where you saw the issue?

Not being Paul but to add some further data points which might be useful:

We have two different clusters running the same kernel version (4.13) + OS (Debian/stretch).

On the cluster where we're seeing this checksum error behavior quite regularly we're running far closer to memory limits:

```
synpromika@virt3 ~ % grep -e MemTotal -e MemFree /proc/meminfo
MemTotal:       65250740 kB
MemFree:         4735408 kB
synpromika@virt3 ~ % free -m

              total        used        free      shared  buff/cache   available
Mem:          63721       45817        3907         680       13996       20484
Swap:             0           0           0
```

We never used swap there, though enabled swap recently (**after** being aware of this issue already) for testing to see whether this makes any changes - though it doesn't seem so, still running into the issue more or less in the same frequency.
FTR, this is the cluster where we have 18 OSDs using Filestore and 8 OSDs with Bluestore.

While on the other cluster (all 9 present OSDs running Bluestore) we **never** saw this error happening, the memory situation is *far* more relaxed there (never used swap there too):

```
synpromika@alpha02 ~ % free -m
              total        used        free      shared  buff/cache   available
Mem:          63595       16174         512          62       46908       48809
Swap:             0           0           0
```

We have a memory upgrade scheduled for the cluster where we're running into this issue, once the memory upgrade has been done I can report back whether we have any changes regarding the checksum error situation.

#41 - 04/27/2018 11:48 AM - Paul Emmerich

Oh, I just remembered something:

We did reduce the Bluestore cache size from 2 GB to 1 GB at around the same time when the problem disappeared because we were starting to run out of memory during recovery/backfilling.

#42 - 04/28/2018 08:26 PM - Paul Emmerich

I've managed to reproduce this on a test cluster but it's somewhat unreliable and took a few attempts.

1. fill test cluster with data
2. increase bluestore cache size to an unreasonably high value (2.4 GB cache size * 6 OSDs on a server with 16 GB RAM)
3. read data with a client
4. wait until a few OSDs are getting killed by the OOM killed
5. issue deep-scrub commands to all OSDs
6. got > 300 scrub errors within a few minutes, this happens on both OSDs that got killed by the OOM killer and ones running continuously

```
root@ct-6-02C4AE ~ $ free -h
              total        used        free      shared  buff/cache   available
Mem:            15G         14G        146M         41M        558M         30M
Swap:            0B          0B          0B
```

```
root@ct-6-02C4AE ~ $ ceph -v
ceph version 12.2.5 (cad919881333ac92274171586c827e01f554a70a) luminous (stable)
root@ct-6-02C4AE ~ $ uname -a
Linux ct-6-02C4AE 4.15.0-0.bpo.2-amd64 #1 SMP Debian 4.15.11-1~bpo9+1 (2018-04-07) x86_64 GNU/Linux
root@ct-6-02C4AE ~ $ lsb_release -a
No LSB modules are available.
Distributor ID:    Debian
Description:    Debian GNU/Linux 9.4 (stretch)
Release:    9.4
Codename:    stretch
```

I wonder if this can be reproduced with a simple tool that just swallows memory instead of increasing the cache size.

I unfortunately can't keep the system in that state for long.
Keeping up the client IO makes working on the system almost impossible or crashes the server (live images really don't like that) and without client IO memory gets back to sane levels after an OOM kill.

The bug disappeared once available memory got back to ~2GB after an OOM kill and the errors disappear. The production cluster managed to stay in this state for months and it wasn't particularly unstable.

#43 - 04/30/2018 10:09 AM - Dennis Björklund

FWIW, I got this error with checksum 0x6706be76 on 12.2.5. I upgraded a couple of days ago and the bug is still there (not surprising).

I'm also on bluestore and I get these errors on a specific node that only have 8GB of memory and the others have 16GB. It has swap and uses it but it has never run out of swap and the OOM-killer hasn't killed anyone.

#44 - 04/30/2018 11:18 AM - Dennis Björklund

Thank Brian for the hint to rerun the deep scrub on the broken pg. It worked fine!

Previously I've been doing repairs and checking for corruption by comparing to backup data (not on ceph). I think it has worked all the times before but it feels much better to rerun the scrub and have it detect that everything actually is in order (which I hope it mean when it doesn't detect anything the second time around).

Does it do the scrubbing on data that hasn't been fully written out yet? How can the next scrub not find an error otherwise?

I run on debian stretch with kernel: SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64 GNU/Linux

#45 - 05/15/2018 10:06 PM - Michael Prokop

Michael Prokop wrote:

> We have a memory upgrade scheduled for the cluster where we're running into this issue, once the memory upgrade has been done I can report back whether we have any changes regarding the checksum error situation.

We did the memory/RAM upgrade on 2018-05-02 and since then we never saw this issue appear again (not a single time).

JFTR, new situation on one of the nodes:

```
synpromika@virt3 ~ % grep -e MemTotal -e MemFree /proc/meminfo
MemTotal:       131310672 kB
MemFree:         1054616 kB
synpromika@virt3 ~ % free -m
              total        used        free      shared  buff/cache   available
Mem:          128233       45815        1252         553       81165       85584
Swap:          16383         392       15991
```

#46 - 05/18/2018 09:58 AM - Emmanuel Lacour

Hi, seems we have same problem here. We just start using a new cluster and aw already 3 scrub errors in one week, always on same osd/node, but without any hardware error reported. A pg repair always fix this.

Our setup:

Debian 9

1. uname -a
   Linux osd-03 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64 GNU/Linux

1. ceph --version
   ceph version 12.2.5 (cad919881333ac92274171586c827e01f554a70a) luminous (stable)

1. zgrep ERR /var/log/ceph/ceph-osd.*
   /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:09.277426 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb shard 3: soid 1:d3a5241a:::rbd_data.59cc72ae8944a.0000000000004c03:head candidate had a read error
   /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:13.556818 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb deep-scrub 0 missing, 1 inconsistent objects
   /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:13.556832 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb deep-scrub 1 errors
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:42.499539 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 shard 3: soid 1:4333e234:::rbd_data.59cc72ae8944a.0000000000005603:head candidate had a read error
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:46.135521 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 deep-scrub 0 missing, 1 inconsistent objects
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:46.135530 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 deep-scrub 1 errors
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:17.438761 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 shard 3: soid 1:4333e234:::rbd_data.59cc72ae8944a.0000000000005603:head candidate had a read error
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:20.946934 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 repair 0 missing, 1 inconsistent objects
   /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:20.946965 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 repair 1 errors, 1 fixed
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:03.978653 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 shard 3: soid 1:858b20bb:::rbd_data.59cc72ae8944a.0000000000004406:head candidate had a read error
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:06.020035 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 deep-scrub 0 missing, 1 inconsistent objects
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:06.020046 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 deep-scrub 1 errors
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:21.643544 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 shard 3: soid 1:858b20bb:::rbd_data.59cc72ae8944a.0000000000004406:head candidate had a read error
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:25.163548 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 repair 0 missing, 1 inconsistent objects
   /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:25.163576 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 repair 1 errors, 1 fixed

1. free -h

	total	used	free	shared	buff/cache	available
Mem:	125G	25G	96G	361M	3.5G	98G
Swap:	3.7G	0B	3.7G			

#47 - 05/18/2018 12:16 PM - Emmanuel Lacour

Emmanuel Lacour wrote:

> Hi, seems we have same problem here. We just start using a new cluster and aw already 3 scrub errors in one week, always on same osd/node, but without any hardware error reported. A pg repair always fix this.

one more time on another osd.

More about our hardware maybe, We used Supermicro servers, SSD for OS and DB, sata disks for OSDs, on raid controller (raid 0 single disk) with battery backed write cache enabled.

c0 | AVAGO MegaRAID SAS 9361-8i | 1024MB | 79C | Good | FW: 24.21.0-0025

We have clusters using ceph hammer with same hardware without problem.

> Our setup:
>
> Debian 9
>
> 1. uname -a
> Linux osd-03 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64 GNU/Linux
>
> 1. ceph --version
> ceph version 12.2.5 (cad919881333ac92274171586c827e01f554a70a) luminous (stable)
>
> 1. zgrep ERR /var/log/ceph/ceph-osd.*
> /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:09.277426 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb shard 3: soid 1:d3a5241a:::rbd_data.59cc72ae8944a.0000000000004c03:head candidate had a read error
> /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:13.556818 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb deep-scrub 0 missing, 1 inconsistent objects
> /var/log/ceph/ceph-osd.8.log:2018-05-18 11:15:13.556832 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.1cb deep-scrub 1 errors
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:42.499539 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 shard 3: soid 1:4333e234:::rbd_data.59cc72ae8944a.0000000000005603:head candidate had a read error
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:46.135521 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 deep-scrub 0 missing, 1 inconsistent objects
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 17:59:46.135530 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 deep-scrub 1 errors
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:17.438761 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 shard 3: soid 1:4333e234:::rbd_data.59cc72ae8944a.0000000000005603:head candidate had a read error
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:20.946934 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 repair 0 missing, 1 inconsistent objects
> /var/log/ceph/ceph-osd.8.log.3.gz:2018-05-15 18:03:20.946965 7f045a886700 -1 log_channel(cluster) log [ERR] : 1.c2 repair 1 errors, 1 fixed
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:03.978653 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 shard 3: soid 1:858b20bb:::rbd_data.59cc72ae8944a.0000000000004406:head candidate had a read error
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:06.020035 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 deep-scrub 0 missing, 1 inconsistent objects
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 11:54:06.020046 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 deep-scrub 1 errors
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:21.643544 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 shard 3: soid 1:858b20bb:::rbd_data.59cc72ae8944a.0000000000004406:head candidate had a read error
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:25.163548 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 repair 0 missing, 1 inconsistent objects
> /var/log/ceph/ceph-osd.9.log.4.gz:2018-05-14 14:18:25.163576 7f9acb7fc700 -1 log_channel(cluster) log [ERR] : 1.1a1 repair 1 errors, 1 fixed
>
> 1. free -h
> total used free shared buff/cache available
> Mem: 125G 25G 96G 361M 3.5G 98G
> Swap: 3.7G 0B 3.7G

#48 - 06/16/2018 05:58 AM - Dennis Björklund

I lowered the memory usage 6 weeks ago by setting "bluestore cache size hdd" to a lower value and after that I've had 0 issues.

#49 - 07/21/2018 08:52 AM - Nathan Cutler

- Related to Bug #25006: bad csum during upgrade test added

#50 - 07/25/2018 09:30 PM - Sage Weil

- Status changed from Verified to Won't Fix

I'm going to close this given that all of the evidence seems to point to a kernel bug with swap.

#51 - 07/26/2018 03:17 PM - Paul Emmerich

I agree that it is cleary a kernel bug in 4.9+, but I disagree with won't fix as a conclusion. Also, it also happens when swap is disabled.

The main problem here is that it can be hard to explain that a cluster suddenly reporting HEALTH_ERROR with > 200 occurences of "possible data damage" is nothing to worry about and that we will just reduce memory usage to fix that.
Users don't like seeing messages about "possible data damage" just because they used too much memory.
This will also bite you in the future if it happens to one of your enterprise customers as soon as RHEL uses a newer kernel.

So the be constructive here: we should be able to create a work-around that mitigates this as much as possible. There is a patch floating around somewhere in this thread that retries the read if a CRC corruption is detected. I had tested this a few months ago and the retried read succeeded almost always.

Maybe this could be a suitable work-around:
If the checksum validation fails and the read was all zeroes: re-try the read up to n times before reporting an error. That should catch almost all of these spurious errors. Real CRC errors perform some useless addtional work but that shouldn't matter since its really rare and a few additional reads are the least of your worries there.

#52 - 07/26/2018 07:37 PM - Paul Emmerich

I've prototyped a work-around here: https://github.com/ceph/ceph/pull/2327

Is there a good reason to not retry reads in this scenario?

#53 - 08/08/2018 10:01 AM - Honggang Yang

Paul Emmerich wrote:

> I've prototyped a work-around here: https://github.com/ceph/ceph/pull/2327
>
> Is there a good reason to not retry reads in this scenario?

https://github.com/ceph/ceph/pull/23273

:)

#54 - 08/09/2018 12:11 PM - Paul Emmerich

Oh, yeah, looks like I fail at copy & pasting URLs. Correct link is https://github.com/ceph/ceph/pull/23273

#55 - 08/22/2018 08:25 AM - Stefan Seidel

Same problem here. I/O blocking on krbd when this bug hits, which needs a reboot of the VM using the KRBD device, and also the host, because not even sync will complete on the host, because the krbd device is 100% busy (with zero IOPS).

```
2018-08-21 22:28:37.298133 7f9fc6eae700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000
checksum at blob offset 0x3000, got 0x6706be76, expected 0x77722c59, device location [0x40a89bb000~1000], logi
cal extent 0x233000~1000, object #1:d41fdeea:::rbd_data.6753fa2ae8944a.0000000000008b5c:head#
```

The terrible thing is that the next message is

```
2018-08-21 22:28:37.299809 7f9fc6eae700 -1 log_channel(cluster) log [ERR] : 1.2b missing primary copy of 1:d41
fdeea:::rbd_data.6753fa2ae8944a.0000000000008b5c:head, will try copies on 12,14
```

which would be fine - **if it would work**. I do wonder why it says it would read from one of the remaining copies, but I/O still stops. I did a deep-scrub of pg 1.2b and it came out clean.

MemFree on the host was 1.2GB around that time, evidenced by the monitoring. MemAvailable around 41.8 GB.

And of course, if this alleged kernel bug would be fixed, that would be good as well. Still I think that it's not unwise to not have a single OSDs I/O failure result in a total blocking I/O on the client.

This is happening to us on a 5 host Cluster, running Debian 9, ceph versions:

```
{
    "mon": {
        "ceph version 12.2.7 (94ce186ac93bb28c3c444bccfefb8a31eb0748e4) luminous (stable)": 5
    },
    "mgr": {
        "ceph version 12.2.7 (94ce186ac93bb28c3c444bccfefb8a31eb0748e4) luminous (stable)": 5
    },
    "osd": {
        "ceph version 12.2.7 (94ce186ac93bb28c3c444bccfefb8a31eb0748e4) luminous (stable)": 14
    },
    "mds": {},
    "overall": {
        "ceph version 12.2.7 (94ce186ac93bb28c3c444bccfefb8a31eb0748e4) luminous (stable)": 24
    }
}
```

#56 - 08/22/2018 12:20 PM - Paul Emmerich

This seems unrelated: retried reads succeed with this bug.

#57 - 08/23/2018 06:42 AM - Stefan Seidel

Paul Emmerich wrote:

> This seems unrelated: retried reads succeed with this bug.

Do you mean that because it says will try copies on 12,14 this is not the same bug?

Because it looked very similar, _verify_csum bad crc32c/0x1000 checksum at blob offset 0x3000, got 0x6706be76. Or did I misunderstand?

Wanted to add kernel version, too: 4.15.18-17.

#58 - 08/23/2018 08:47 AM - Paul Emmerich

Not saying that it's completely unrelated, but check out http://tracker.ceph.com/issues/24901

#59 - 09/04/2018 01:06 PM - Alfredo Rezinovsky

Dennis Björklund wrote:

> I lowered the memory usage 6 weeks ago by setting "bluestore cache size hdd" to a lower value and after that I've had 0 issues.

How much lower ?

I have 3 OSDs per host (16 Gb RAM) and still have the problem with 13.2.1 (5533ecdc0fda920179d7ad84e0aa65a127b20d77) mimic (stable)

#60 - 09/04/2018 06:21 PM - Dennis Björklund

I've run it with

bluestore cache size hdd = 134217728

and it still hasn't happened even once after I changed. Before it happened every week.

#61 - 10/03/2018 09:56 PM - Nick Fisk

I've just bit hit by a wave of these after upgrading to Mimic, everything else remains the same, no reboot was carried out.

My working theory is that this is possibly not directly related to swap but lack of free (not available) memory, swapping would probably just be one

symptom of having low free memory. I realised that all the nodes I was seeing this on were missing the "vm.min_free_kbytes = 4194304" I normally have. The nodes that were experiencing these read errors had 30+GB available but less than 200MB free.

Is this possibly some form of allocation error as the memory is in use by the linux buffer/cache and doesn't release it in time/properly???

Anyway will report back in the next 48 hours or so, if the free memory sysctl tuning has helped.

#62 - 10/04/2018 08:01 PM - Paul Emmerich

Update from our side: we've been running the patch I've posted above since Mimic for all our production clusters. No problems since, works perfectly.

#63 - 10/10/2018 09:06 AM - Nick Fisk

Reporting back, increasing min_free_kbytes has not appeared to have helped. Swap usage is only a couple of MB out of 2GB.

This is on stock Ubuntu 18.04 kernel btw.

#64 - 10/17/2018 06:31 PM - Jan Pekař

And can you specify, which kernel issue/bug you are talking about?. You mentioned 4.9+ kernel problem. Do you have any link somewhere?
I think it is pretty serious.
From that time (looks like from kernel upgrade) I'm detecting problem on my testing cluster which is short on memory and it is swapping a lot when rebuilding. I detected saved and acknowledged files on cephfs erasure pool with one rados object empty (all zeroes) inside the file (hard to replicate this issue now). Also detecting unfound objects after major cluster reweight (after adding disks etc). It can be another issue, but now it looks to me, that is all connected together with this.

Are you sure, that this is only happening during scrub and it is minor issue / wontfix issue? What happens, when this kernel? issue occur during recovery/rebuild?

I'm using erasure 3+1 so maybe when this issue hits 2 blocks during recovery, I'm lost with my block.

I know, that for production I should use erasure +2 or 3 replicas, but it is not excuse, if everything what I wrote is true and one single block is in danger.

#65 - 10/22/2018 09:00 AM - Gaudenz Steinlin

We are hitting this bug as well. In our cluster it occurred 14 times in the last 50 days.

This is our setup:

- 3 Node cluster Running Proxmox
- 4 HDD OSDs on each node
- 2 SSD OSD on each node
- All OSDs built with bluestore, HDD OSDs have their WAL and block.db on an SSD (same SSD for all 4 OSDs)
- Ceph version 12.2.8-pve1 built by Proxmox
- 1 Pool with CRUSH rule to limit to SSDs, 1 Pool with CRUSH rule to limit to HDDs, CephFS pools with CRUSH rule to limit to HDDs

The scrub errors with checksum mismatch matching "0x6706be76" only happen on the HDD OSDs. Just issuing a "ceph pg deep-scrub" on the affected PG "solves" the problem.

We first had Swap activated on these nodes, but as we don't really need Swap we deactivated it. This DID NOT change anything. The problem occurs in about the same frequency. We then tried to "bluestore cache size hdd = 134217728" which seems to make the problem occur less frequent, but it still occurs about once in 4 days.

In this cluster we first upgraded the HDD OSDs to bluestore and only upgraded the SSD OSDs to bluestore about 3 weeks later. The problem started as soon as the HDD OSDs were converted. Going back over the history of srub errors I noticed that the problem occurred much more frequently (about once every second day) after we upgraded the SSD OSDs to bluestore. The frequency went down again when we changed the cache size about 2 weeks later. This might just be a coincidence.

I'm willing to help debug this a much as possible. Please tell me if you need more data or have any ideas about other things to try out. As multiple people have now reported that this also happens with Swap deactivated, I don't think this is caused by swapping. Our performance data does not show any memory pressure at the moment the scrub errors happen.

#66 - 10/23/2018 04:48 PM - Nick Fisk

I think I maybe seeing this on actual client requests as well as scrubs. Since upgrading to Mimic and these scrub errors have started happening I've noticed that a client with several KRBD mounts has occasionally started IO hanging and requires a hard reset to get full IO working again. First time I didn't think much of it, but it's happened a few more times now.

Checking the kernel log on the client I see this at exactly the same time as the CPU load jumps up and sticks

Oct 20 03:30:02 MS-CEPH-Proxy3 kernel: [155528.899503] libceph: get_reply osd59 tid 42415883 data 4194304 > preallocated 49152, skipping

Interestingly, nothing on OSD59, so not sure how that OSD number relates to the errors below

ceph.log reports at same time:
2018-10-20 03:30:02.066427 osd.20 osd.20 10.3.31.12:6813/2942352 32 : cluster [ERR] 0.3ba missing primary copy of
0:5dc5bd1d:::rbd_data.1555406238e1f29.00000000002a3025:
head, will try copies on 54,66

osd.20 log:
2018-10-20 03:30:01.923 7f25878df700 -1 bluestore(/var/lib/ceph/osd/ceph-20) _verify_csum bad crc32c/0x1000 checksum at blob offset 0xb000, got 0x6
706be76, expected 0x1207716b, device location [0x24f7950b000~1000], logical extent 0xb000~1000, object
#0:5dc5bd1d:::rbd_data.1555406238e1f29.00000
000002a3025:head#
2018-10-20 03:30:02.059 7f25878df700 -1 log_channel(cluster) log [ERR] : 0.3ba missing primary copy of
0:5dc5bd1d:::rbd_data.1555406238e1f29.000000
00002a3025:head, will try copies on 54,66
2018-10-20 03:30:02.171 7f25878df700 0 osd.20 pg_epoch: 199348 pg[0.3ba(v 199348'7544022 (199340'7540931,199348'7544022] local-lis/les=199201/199
202 n=5000 ec=1/1 lis/c 199201/199201 les/c/f 199202/199202/141272 199201/199201/199144) [20,66,54] r=0 lpr=199201 rops=1 crt=199348'7544022 lcod 1
99348'7544021 mlcod 199348'7544021 active+recovering mbc={255={(3+0)=1}}] _update_calc_stats ml 1 upset size 3 up 3
2018-10-20 03:30:02.171 7f25878df700 0 osd.20 pg_epoch: 199348 pg[0.3ba(v 199348'7544022 (199340'7540931,199348'7544022] local-lis/les=199201/199
202 n=5000 ec=1/1 lis/c 199201/199201 les/c/f 199202/199202/141272 199201/199201/199144) [20,66,54] r=0 lpr=199201 rops=1 crt=199348'7544022 lcod 1
99348'7544021 mlcod 199348'7544021 active+recovering mbc={255={(3+0)=1}}] _update_calc_stats ml 1 upset size 3 up 3
2018-10-20 03:30:02.171 7f25878df700 0 osd.20 pg_epoch: 199348 pg[0.3ba(v 199348'7544022 (199340'7540931,199348'7544022] local-lis/les=199201/199
202 n=5000 ec=1/1 lis/c 199201/199201 les/c/f 199202/199202/141272 199201/199201/199144) [20,66,54] r=0 lpr=199201 rops=1 crt=199348'7544022 lcod 1
99348'7544021 mlcod 199348'7544021 active+recovering mbc={255={(3+0)=1}}] _update_calc_stats ml 1 upset size 3 up 3

Nothing in either of 54 or 66 osd logs.

#67 - 10/24/2018 12:20 AM - Mark Lopez

Odd, I got the same error Nick.

> libceph: get_reply osd4 tid 1850429 data 1835008 > preallocated 262144, skipping

But I think it was related to a bug in CephFS - it's causing uninterruptible processes via "Leaked POSIX lock":

> BUG: unable to handle kernel paging request at ffffffffffffff24
> IP: ceph_fl_release_lock+0x14/0x50 [ceph]

I can only seem to correlate the checksum errors with scrubs - since they're scheduled for nighttime executions. There's a couple scrub errors every night, but it is a small cluster - only 24TB of data.

The Ceph cluster and the clients are both running Ubuntu LTS 18.04 - stock kernel. Zero bytes of swap is in use - even during scrubbing.

#68 - 10/29/2018 08:10 PM - Yuri Weinstein

https://github.com/ceph/ceph/pull/24647 merged

#69 - 11/29/2018 01:08 AM - Yuri Weinstein

https://github.com/ceph/ceph/pull/24649 mergedhttps://github.com/ceph/ceph/pull/24649

#70 - 01/07/2019 10:02 AM - Gaudenz Steinlin

The upgrade to 12.2.10 fixed the issue for us. See #65 for our setup. The most likely change in 12.2.10 which fixed this is commit a7bcb26023e90bcd7409d5dbe5fea72afcfa8e61.

#71 - 02/11/2019 10:06 AM - Nick Fisk

I still seem to be experiencing these errors, albeit at a much reduced rate since upgrading to 13.2.3. I could wake up in the morning and find 20+ inconsistent PG's, now its normally around 2-3. I'm still also experiencing random IO hang's, I'm trying to tally if these are still related.

Have there been any developments from a kernel perspective which addresses this problem at the source?

2019-02-11 08:05:07.781 7f1026512700 0 log_channel(cluster) log [DBG] : 0.a87 deep-scrub starts
2019-02-11 08:14:49.504 7f1026512700 0 log_channel(cluster) log [DBG] : 0.a87 deep-scrub ok
2019-02-11 08:14:56.620 7f1025510700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x41340bf1, device location [0x1bf60970000~1000], logical extent 0x0~1000, object #0:0ae03130:::rbd_data.1555406238e1f29.00000000000aca9f:head#
2019-02-11 08:17:08.930 7f1025510700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0xde80428f, device location [0x1e8fb340000~1000], logical extent 0x80000~1000, object #0:0ae391ad:::rbd_data.158f204238e1f29.00000000000f6a96:head#
2019-02-11 08:17:43.238 7f1025510700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _verify_csum bad crc32c/0x1000 checksum at blob offset 0x0, got 0x6706be76, expected 0x6b86f6a1, device location [0x1986f690000~1000], logical extent 0x180000~1000, object #0:0ae48795:::rbd_data.1555406238e1f29.0000000000361dd2:head#
2019-02-11 08:17:54.098 7f1025510700 -1 bluestore(/var/lib/ceph/osd/ceph-1) _decompress decompression failed with exit code -2

2019-02-11 08:17:54.126 7f1025510700 -1 /build/ceph-13.2.3/src/os/bluestore/bluestore_types.h: In function 'static void bluestore_compression_header_t::_denc_finish(ceph::buffer::ptr::iterator&, __u8*, __u8*, char**, uint32_t*)' thread 7f1025510700 time 2019-02-11 08:17:54.109566
/build/ceph-13.2.3/src/os/bluestore/bluestore_types.h: 1016: FAILED assert(pos <= end)

Files

ceph-errors	5.95 KB	01/19/2018	Martin Preuss