RADOS - Bug #21174

OSD crash: 903: FAILED assert(objiter->second->version > last_divergent_update)

08/30/2017 06:12 AM - Martin Millner

Status: Rejected
Priority: Urgent
Assignee:
Category:
Target version: v12.2.0
Source: Community (user)
Tags:
Backport:
Regression: No
Severity: 2 - major
Reviewed:

Affected Versions:
ceph-qa-suite:
Component(RADOS):

Description
I've setup a cephfs erasure coded pool on a small cluster consisting of 5 bluestore OSDs. The pools were created as follows:

```
ceph osd pool create cephfs_metadata 160 160 replicated
ceph osd pool create cephfs_data 160 160 erasure ecpool ec
ceph osd pool set cephfs_data allow_ec_overwrites true
ceph fs new cephfs cephfs_metadata cephfs_data
```

I started copying files onto the cephfs, which has now started to crash in endless loops. Cluster is unavailable (which is uncritical for me but not for a "live cluster"). The log which is available at [https://martin.millnert.se/files/cephfs_ec/ceph-osd.1.log.all.gz](https://martin.millnert.se/files/cephfs_ec/ceph-osd.1.log.all.gz) crashes in the vicinity of a lot of output about "omap" operations.

In the documentation at [http://docs.ceph.com/docs/master/rados/operations/erasure-code/](http://docs.ceph.com/docs/master/rados/operations/erasure-code/) it is stated that erasure coded pools do not support omap operations, which is why special care has to be taken with RBD. For CephFS, it simply states: "For Cephfs, using an erasure coded pool means setting that pool in a file layout." with a link to the section on CephFS file layouts: [http://docs.ceph.com/docs/master/cephfs/file-layouts/](http://docs.ceph.com/docs/master/cephfs/file-layouts/)
The file layouts documentation section does not reciprocate this link, i.e. the logic/context of using erasure coded pools with cephfs is not further explained there.

So, provided I've done a user error here and there is no other bug causing my OSDs to crash, I think it would be wise to upgrade the documentation on how to use EC pools for CephFS more explicitly.

Furthermore, if it is indeed illegal to create a cephfs using the command I did, i.e. "ceph fs new cephfs <replicated_metadata_pool> <erasure_coded_data_pools>", probably the code should test and reject that to avoid cluster down states further down the road.

Related issues:
Resolved
Closed

History
#1 - 08/30/2017 11:25 AM - John Spray
- Subject changed from CephFS erasure coded pool not explained in documentation to OSD crash using EC pool for CephFS
06/05/2021
Subject changed from OSD crash using EC pool for CephFS to OSD crash: 903: FAILED assert(objiter->second->version > last_divergent_update)

Martin: just to confirm, you were seeing this crash while you had EC pools involved, and when you do not have any EC pools you do not see the crash?

To clarify then: I have not tested this with a replicated cephfs data pool. Only tested with ec data pool as per my 4 commands above.

I can do one of two things:
1. I can scrap the cephfs fs and the cephfs ec pool (cephfs_data above) and recreate it with a replicated pool. There is no data loss if I scrap the fs and start over.
2. I can wait out for some hopeful fix and not have to scrap ~2 days worth of IO.

If you want me to do 1) for double verification of replicated, I can do that. Just let me know.

I've got exactly the same problem with kernel client. But fuse client seems fine with ec pool on cephfs

duplicate issue: http://tracker.ceph.com/issues/16279

- Project changed from Ceph to RADOS
- Category deleted (OSD)
I have a similar issue with OSDs dropping out:

```
```

ceph-post-file: 3c7be3d8-dea9-4cf9-91ef-470e81d1b29f

---

#8 - 03/17/2019 12:10 AM - Sage Weil

- Status changed from New to 12
- Priority changed from Normal to Urgent

---

```
-103> 2019-03-16 23:01:20.667 7fd576365700 10 osd.5 pg_epoch: 65479 pg[1.96s3( v 5163'138877 lc 3142'138872 (2469'135808,5163'138877] local-lis/les=5371/5372 n=65596 ec=71/51 lis/c/f 65428/5139/0 65477/65469) [4, 6, 5, 2147483647, 1) p4(0) r=3 lpr=65477 pi=[5139, 65477]/24 crt=5163'138877 lcod 0'0 unknown NO TIFY m=2 mbc={}] state=Started/Stray: got info+log from osd.4(0) 1.96s0( v 7471'138879 (2469'135808,7471'138879] local-lis/les=65477/65479 n=66596 ec=71/51 lis/c/f 65428/5139/0 65477/65469) log((5162'138874,7471'138879], crt=7471'138879)

-102> 2019-03-16 23:01:20.667 7fd576365700 10 merge_log log((5162'138874,7471'138879], cr
```
For completeness: The root cause for the crashes I experienced were that I had oversized RADOS objects (2-10GB, max recommended is in the <100 MBs or so (see docs for proper answer)) in my non-erasure coded, non-BlueStore RADOS objects. EC-pool on BlueStore (not clear on the exact relation here) enforced strict bounds on RADOS object sizes.

I worked around the issue by setting up cephfs and migrating out the objects onto that instead.

Err. I believe I mixed up two different bugs, please disregard my previous comment. I don't currently recall what I did to solve the above on my end.

Looking at the logs, it seems that the first crash was seen on osd.2 on pg id 1.cas2

before the crash, this is what happened

FWIW, the crash seems similar to https://tracker.ceph.com/issues/36598, but could have a totally different root cause, which resulted in the object not being present.

The failure on osd.5 is seen afterwards on pg id 1.96s3

This happens while doing the following.

when it finds
and asserts because objiter->second->version(5338'138875) is equal to last_divergent_update: 5338'138875.

I have more investigation to do on this, but it will be useful to have the logs from osd.0 and osd.4, which were the corresponding primaries.

---

**#12 - 03/20/2019 10:41 AM - Grant Slater**

As requested.

osd.0: ceph-post-file: 17efe900-501c-479f-ba56-dd29fef18c58
osd.4: ceph-post-file: ff22f830-e6bc-4ffc-a051-3c3145ac638d

---

**#13 - 03/22/2019 01:33 AM - Neha Ojha**

It is possible that the crash we are seeing on osd.2 is due to 1:537949df:::20000a2c834.00000105:head incorrectly rolling back in Case 5 of _merge_object_divergent_entries().

if (!i->can_rollback() || i->version <= olog_can_rollback_to) decides whether we can_rollback or not, i->version here is 5163'140311, but looking at the logs it is hard to find out what the value of olog_can_rollback_to was. All we know is that we decided to rollback and after merge_log() finished, crt=5281'140312.

We might need to preserve the original crt on the merge target, something like what we were trying to do in https://github.com/ceph/ceph/pull/26347. But, it is hard to tell without knowing more details.

I have a PR that explicitly prints olog_can_rollback_to https://github.com/ceph/ceph/pull/27105, before making that decision.

Grant, how easily can you reproduce this? Would it be possible for you to apply a patch and reproduce this failure?

---

**#14 - 03/24/2019 04:08 PM - Grant Slater**

Yes I can still reproduce it, the cluster is still in a broken state.

```
cluster:
  id: 08ff77c3-5f4c-4757-afc8-56ea67bf4343
  health: HEALTH_WARN
  1 filesystem is degraded
    noout,noscrub,nodeep-scrub flag(s) set
    2 osds down
    Reduced data availability: 134 pgs inactive, 47 pgs down, 81 pgs incomplete
    Degraded data redundancy: 18090022/209172759 objects degraded (8.648%), 295 pgs degraded, 295 pgs undersized

  services:
    mon: 5 daemons, quorum cluster-01,cluster-02,cluster-03,cluster-04,cluster-05
    mgr: cluster-02(active), standbys: cluster-05, cluster-03, cluster-01, cluster-04
    mds: cephfs-4/4/4 up {0=cluster-03=up:replay,1=cluster-01=up:resolve,2=cluster-02=up:resolve,3=cluster-05=up:resolve}, 1 up:standby
    osd: 24 osds: 22 up, 24 in; 9 remapped pgs
      flags noout,noscrub,nodeep-scrub
```

06/05/2021 8/13
I have built and deployed with the [https://github.com/ceph/ceph/pull/27105](https://github.com/ceph/ceph/pull/27105) logging patch.

Attached are the logs for the 2x OSD that mainly crash:

- osd.2: ceph-post-file: 239e8b8c-70b6-479a-a822-f2813b1b9991
- osd.5: ceph-post-file: 7c843140-42d1-4cda-a9ba-a04354fe5fca

Attached are the logs for the other OSD that occasionally crash:

- osd.0: ceph-post-file: ffa15979-c3be-44cc-b563-b4f6a72ed2b2d
- osd.1: ceph-post-file: 5bd6bf20-f9e6-4b27-8acd-1f374a761b2a
- osd.4: ceph-post-file: b65b6b2e-5500-48a7-83cf-5ec727b69aa

---

**#15 - 03/28/2019 02:23 AM - Neha Ojha**

[https://github.com/ceph/ceph/pull/27200](https://github.com/ceph/ceph/pull/27200) attempts to resolve the failure seen on osd.2

**#16 - 03/28/2019 05:22 PM - Grant Slater**

I am still seeing crashes with [https://github.com/ceph/ceph/pull/27200](https://github.com/ceph/ceph/pull/27200) backported.

Attached are logs:

- osd.2 ceph-post-file: fb80a2a5-e373-42e4-ad30-27121b2a5777
- osd.5 ceph-post-file: 4b6074c-8480-47a3-a759-b2729b5d0e87
Hi Grant,

Thanks for applying the patch and updating the logs. Looks like the earlier crash on osd.2 (ENOENT on clone suggests osd bug) has now gone away, and now both osds are reporting the same problem. Will investigate this further.

osd 2

```
-9> 2019-03-28 17:14:46.482 7f03e9ce7700 20 osd.2 pg_epoch: 66018 pg[2.3a] (v 65910/174237 (43944/171225, 65910/174237) local-lis/les=66017/66018 =m16265 ecc=139/139 lis/c=66015/65471 les/c=66016/65473/0 66017/66017/66017 17) [2,3,4] r=0 lpr=66017 pi=[65471,66017]/3 cct=65910/174237 lcod 0'0 micd 0'0 activating mbc={}) rollforward: entry=43947/172084 (1466/48358) modify 2:5f7e2597::200007988b5.000000000 head by osd.3.0:38024 0.000000 0
-9> 2019-03-28 17:14:46.482 7f03e94e6700 -1 [src/osd/PGLog.h: In function "static void PGLog::_merge_object_divergent_entries(const PGLog::IndexedLog&, const hobject_t&, mempool::osd_pglog::list<pg_log_entry_t>&, const pg_info_t&, eversion_t, eversion_t, missing_type&, PGLog::LogEntryHandler*, const DoutPrefixProvider*) [with missing_type = pg_missing_set<true>; mempool::osd_pglog::list<pg_log_entry_t> = std::__cxx11::list<pg_log_entry_t, mempool::pool_allocator<(mempool::pool_index_t)14, pg_log_entry_t> >] thread 7f03e94e6700 0 time 2019-03-28 17:14:46.484179
```

cph version 13.2.5 (chff874f9007f1869bfdf3821b7e33b2a6f4df4988) mimic (stable)

1: (ceph::cceph::assert_fail(char const*, char const*, int, char const*)+0x102) [0x7f040cabb5c2]
2: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x102) [0x7f040cabb787]
3: (void PGLog::_merge_object_divergent_entries<pg_missing_set<true> &>::merge_process(const PGLog::IndexedLog const&, const hobject_t&, mempool::osd_pglog::list<pg_log_entry_t>&, const pg_info_t&, eversion_t, eversion_t, pg_missing_set<true>&, PGLog::LogEntryHandler*, const DoutPrefixProvider*) [with missing_type = pg_missing_set<true>; mempool::osd_pglog::list<pg_log_entry_t> = std::__cxx11::list<pg_log_entry_t, mempool::pool_allocator<(mempool::pool_index_t)14, pg_log_entry_t> >] thread 7f03e94e6700 0 time 2019-03-28 17:14:46.484179

osd.5

```
-1> 2019-03-28 17:03:25.077 7f878c5cd700 20 osd.2 pg_epoch: 66005 pg[1.cas1] (v 5281/140312 (2469/137238, 5281/140312) local-lis/les=66000/66003 =m5281/140312 ecc=71/51 lis/c=66001/5135 les/c=66001/5144/0 66000/66000/65994) [0,5,2147483647,3,4,7]p0(0) r=1 lpr=66000 pi=[5135,66000]/18 luod=0'0 cct=5281/140312 lcod 0'0 active mbc={}) handle_activate_map: Not dirtying info: last_persisted is 66003 while current is 66005
0> 2019-03-28 17:03:25.077 7f878d5cf700 -1 [src/osd/PGLog.h: In function "static void PGLog::_merge_object_divergent_entries(const PGLog::IndexedLog&, const hobject_t&, mempool::osd_pglog::list<pg_log_entry_t>&, const pg_info_t&, eversion_t, eversion_t, missing_type&, PGLog::LogEntryHandler*, const DoutPrefixProvider*) [with missing_type = pg_missing_set<true>; mempool::osd_pglog::list<pg_log_entry_t> = std::__cxx11::list<pg_log_entry_t, mempool::pool_allocator<(mempool::pool_index_t)14, pg_log_entry_t> >] thread 7f878d5cf700 0 time 2019-03-28 17:03:25.077113
```

cph version 13.2.5 (chff874f9007f1869bfdf3821b7e33b2a6f4df4988) mimic (stable)

1: (ceph::cceph::assert_fail(char const*, char const*, int, char const*)+0x102) [0x7f040cabb5c2]
2: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x102) [0x7f040cabb787]
3: (void PGLog::_merge_object_divergent_entries<pg_missing_set<true> &>::merge_process(const PGLog::IndexedLog const&, const hobject_t&, mempool::osd_pglog::list<pg_log_entry_t>&, const pg_info_t&, eversion_t, eversion_t, pg_missing_set<true>&, PGLog::LogEntryHandler*, const DoutPrefixProvider*) [with missing_type = pg_missing_set<true>; mempool::osd_pglog::list<pg_log_entry_t> = std::__cxx11::list<pg_log_entry_t, mempool::pool_allocator<(mempool::pool_index_t)14, pg_log_entry_t> >] thread 7f878d5cf700 0 time 2019-03-28 17:03:25.077113
```

06/05/2021 10/13
#18 - 03/28/2019 10:23 PM - Neha Ojha
- Related to Bug #39023: osd/PGLog: preserve original_crt to check rollbackability added

#19 - 04/01/2019 10:46 PM - Samuel Just
Grant: I notice that the initial event outlined above is from October. Is that the very first anomalous behavior exhibited by this cluster? I notice from the osd.2 logs that 1.cas0 was already missing two osds. Were there failures prior to that point?

#20 - 04/02/2019 06:29 PM - Neha Ojha
Hi Grant, is there a way you could dump the pg log by using a command like this "ceph-objectstore-tool --no-mon-config --data-path /dev/osd0 --op log --pgid 1.0" for 1.cas0 to start with?

#21 - 04/02/2019 07:51 PM - Grant Slater
Output from: ceph-objectstore-tool --no-mon-config --data-path /var/lib/ceph/osd/ceph-0 --op log --pgid 1.cas0
1.cas0 ceph-post-file: afac3c51-58c7-497a-9e58-564cb43f4755

#22 - 04/02/2019 10:22 PM - Grant Slater
Per request on irc.
pg log:
1.cas2 on osd.2: ceph-post-file: d74a0006-c0e9-41b1-a904-7bfe41617253
1.96s3 on osd.5: ceph-post-file: 80178cca-1a7b-44b8-9226-df345f53d15c

06/05/2021
- Related to Bug #16279: assert(objiter->second->version > last_divergent_update) failed added

Maybe we could check each log in load_pgs(). If it is corrupt (head != head entry's version), move PG aside and ignore it. If this bug's reporter boots a cluster with this OSD patch maybe the cluster can recover/backfill all the PG shards that were lost.

A less invasive scenario is to report the bad logs and then export-remove all PGs that reported as bad throughout the cluster. This way the bad PGs are saved, but maybe the cluster can recover itself. Send one of the exports using ceph-post-file.

Here is what the bad log looks like that caused one of the crashes. Clearly head is bad because the log ends with 5338'138875 instead of 5163'138877.

Did a corruption occur during 5163 epoch or when a previous merge attempt was made? Or is this a log update not happening due to unwritten data because of crash or power outage?

```

```
#26 - 04/05/2019 12:00 PM - Grant Slater
Yes, most likely the issue was triggered by a power outage, the 2x OSD FAILED assert and the cluster is unable to recover.

#27 - 05/09/2019 02:05 AM - David Zafman
- Status changed from 12 to Rejected

I'm closing this bug. The hardware configuration must make data safe that has been sync'ed to disk. This requires the proper disk cache settings.

An upgrade to Nautilus would also be recommended.