RADOS - Bug #46443

ceph_osd crash in _committed_osd_maps when failed to encode first inc map

07/10/2020 07:42 AM - Markus Binz

Status:	Resolved	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Dan van der Ster		
Category:			
Target version:			
Source:	Community (user)	**Affected Versions:**	v14.2.10, v15.2.1, v15.2.2, v15.2.3, v15.2.4
Tags:		**ceph-qa-suite:**	
Backport:	nautilus, octopus	**Component(RADOS):**	OSD
Regression:	Yes	**Pull request ID:**	36297
Severity:	1 - critical	**Crash signature (v1):**	
Reviewed:		**Crash signature (v2):**	

Description

We upgraded a mimic cluster to v14.2.10, everything was running and ok.
I triggerd an monmap change with the command,
ceph config set global mon_warn_on_pool_pg_num_not_power_of_two false

which resulted in ceph_osd processes crashing (30 out of 50)

later on, it seems to happen on any monmap change. (add osd...)

downgrade to 14.2.9 solved the problem for us.

We have 196 crash reports, i attached just one.

It's the same for 16.04 or 18.04.

{
"crash_id": "2020-06-30_21:27:08.639797Z_a6cf1fdd-5cd6-4355-86d3-bbd39a4d8164",
"timestamp": "2020-06-30 21:27:08.639797Z",
"process_name": "ceph-osd",
"entity_name": "osd.30",
"ceph_version": "14.2.10",
"utsname_hostname": "bigstore06.solnet.ch",
"utsname_sysname": "Linux",
"utsname_release": "4.4.0-101-generic",
"utsname_version": "#124-Ubuntu SMP Fri Nov 10 18:29:59 UTC 2017",
"utsname_machine": "x86_64",
"os_name": "Ubuntu",
"os_id": "ubuntu",
"os_version_id": "16.04",
"os_version": "16.04.6 LTS (Xenial Xerus)",
"backtrace": [
"(()+0x11390) [0x7fcb74ef4390]",
"/usr/bin/ceph-osd() [0x87fd12]",
"(OSD::_committed_osd_maps(unsigned int, unsigned int, MOSDMap*)+0x5e1) [0x8f0f91]",
"(C_OnMapCommit::finish(int)+0x17) [0x946897]",
"(Context::complete(int)+0x9) [0x8fbfb9]",
"(Finisher::finisher_thread_entry()+0x15e) [0xeb2b8e]",
"(()+0x76ba) [0x7fcb74eea6ba]",
"(clone()+0x6d) [0x7fcb744f141d]"
]
}

Related issues:

Related to RADOS - Bug #43903: osd segv in ceph::buffer::v14_2_0::ptr::releas...	**Resolved**
Copied to RADOS - Backport #46741: nautilus: ceph_osd crash in _committed_osd...	**Resolved**
Copied to RADOS - Backport #46742: octopus: ceph_osd crash in _committed_osd_...	**Resolved**

History

#1 - 07/10/2020 09:39 PM - Dan van der Ster

- Subject changed from v14.2.10 Nautilus ceph_osd crash to v14.2.10 Nautilus ceph_osd crash in _committed_osd_maps

fa842716b6dc3b2077e296d388c646f1605568b0 arrived in v14.2.10 and touches _committed_osd_maps

#2 - 07/13/2020 07:21 PM - Neha Ojha

- Project changed from Ceph to RADOS

- Category deleted (OSD)

#3 - 07/16/2020 01:14 PM - Dan van der Ster

Markus do you have a coredump available for further debugging?

#4 - 07/20/2020 01:21 AM - Dan van der Ster

Initially there's a crc error building the full from the first incremental in the loop:

```
 -7726> 2020-06-30 21:27:08.626 7fcb54b0d700 20 osd.30 385679 OSD::ms_dispatch: osd_map(385680..385719 src has
 384938..385811) v4
 -7718> 2020-06-30 21:27:08.626 7fcb54b0d700 20 osd.30 385679 _dispatch 0x25030000 osd_map(385680..385719 src
has 384938..385811) v4
 -7714> 2020-06-30 21:27:08.626 7fcb54b0d700  3 osd.30 385679 handle_osd_map epochs [385680,385719], i have 38
5679, src has [384938,385811]
 -7710> 2020-06-30 21:27:08.626 7fcb54b0d700 10 osd.30 385679 handle_osd_map  got inc map for epoch 385680
 -7145> 2020-06-30 21:27:08.626 7fcb54b0d700  2 osd.30 385679 got incremental 385680 but failed to encode full
 with correct crc; requesting
 -7139> 2020-06-30 21:27:08.626 7fcb54b0d700  0 log_channel(cluster) log [WRN] : failed to encode map e385680
with expected crc
 -1779> 2020-06-30 21:27:08.626 7fcb54b0d700 20 osd.30 385679 my encoded map was:
 -1723> 2020-06-30 21:27:08.634 7fcb54b0d700 10 osd.30 385679 request_full_map 385680..385719, previously requ
ested 0..0
 -1717> 2020-06-30 21:27:08.634 7fcb54b0d700 10 monclient: _send_mon_message to mon.b at v2:10.34.1.11:3300/0
 -1683> 2020-06-30 21:27:08.634 7fcb54b0d700 20 osd.30 385679 handle_osd_map pg_num_history pg_num_history(e38
5679 pg_nums {12={385600=512},56={385666=1}} deleted_pools )
 -1681> 2020-06-30 21:27:08.634 7fcb54b0d700 10 osd.30 385679 write_superblock sb(1d11f1fb-f622-4ab5-98fd-0f53
4400bdf3 osd.30 d02c0656-0cee-4bdd-a5fe-bc768a87c73b e385679 [384938,385679] lci=[342498,385512])
  -472> 2020-06-30 21:27:08.638 7fcb54b0d700 20 osd.30 385679 OSD::ms_dispatch: osd_map(385680..385719 src has
 384938..385811) v4
```

Here's the code:

```
  for (epoch_t e = start; e <= last; e++) {
...
      if ((inc.have_crc && o->get_crc() != inc.full_crc) || injected_failure) {
  dout(2) << "got incremental " << e
    << " but failed to encode full with correct crc; requesting"
    << dendl;
  clog->warn() << "failed to encode map e" << e << " with expected crc";
  dout(20) << "my encoded map was:\n";
  fbl.hexdump(*_dout);
  *_dout << dendl;
  delete o;
  request_full_map(e, last);
  last = e - 1;
  break;
      }
```

So at the beginning we have e = start = 385680 and last = 385719.
But then we request_full_map and set last = 385680-1 = 385679.

And then handle_osd_map doesn't fully bail out -- it continues like it needs to commit from 385680 to 385679.

This makes _committed_osd_maps have last < first:

```
 -1269> 2020-06-30 21:27:08.634 7fcb65364700 10 osd.30 385679 _committed_osd_maps 385680..385679
  -354> 2020-06-30 21:27:08.638 7fcb65364700 -1 *** Caught signal (Segmentation fault) **
 in thread 7fcb65364700 thread_name:fn_odsk_fstore
```

As a result, this whole block in _committed_osd_maps is skipped:

```
  OSDMapRef osdmap;

  // advance through the new maps
  for (epoch_t cur = first; cur <= last; cur++) {
    dout(10) << " advance to epoch " << cur
       << " (<= last " << last
...
```

then I am guessing that the segfault is here just after that loop:

```
  }

  had_map_since = ceph_clock_now();

  epoch_t _bind_epoch = service.get_bind_epoch();
  if (osdmap->is_up(whoami) &&
```

I don't know why the inc crc failed which triggered this. But it seems that the request_full_map logic is broken somehow.

I quickly checked, and indeed this reproduces on any osd with osd_inject_bad_map_crc_probability=1.

#5 - 07/24/2020 06:16 AM - Xiaoxi Chen

I do have coredump captured , the osdmap is null which lead to segmentation fault in osdmap->isup

#6 - 07/24/2020 06:21 AM - Dan van der Ster

@Xiaoxi thanks for confirming. What are the circumstances of your crash? Did it start spontaneously after you upgraded to 14.2.10 like the case of Markus?

#7 - 07/27/2020 06:18 AM - Xiaoxi Chen

@Dan
Yes/no, it is not 100% same that in our case we have several clusters that start adding OSDs with 14.2.10 into the cluster(while others OSDs are still in 2.x and mon are also in 2.2/2.4). Some of the clusters has osds added smoothly but some are not.

For those osd cannot start ,it is 100% reproducible.

#8 - 07/27/2020 06:19 AM - Xiaoxi Chen

Dan van der Ster wrote:

> @Xiaoxi thanks for confirming. What are the circumstances of your crash? Did it start spontaneously after you upgraded to 14.2.10 like the case of Markus?

But the trace and code logic matches with markus's case

#9 - 07/27/2020 07:52 AM - Dan van der Ster

> For those osd cannot start ,it is 100% reproducible.

Could you set debug_ms = 1 on that osd, then inspect the log to see which peer is sending the inc map message which cannot be encoded correctly? We need to know which exact version the peer osd is running.

#10 - 07/27/2020 08:39 AM - Xiaoxi Chen

I think it is mon not the peer OSD. (We just upgrade the mon from 14.2.10 to 15.2.4, below log with mon 15.2.4).

```
2020-07-27 01:36:46.191 7f02e472fc00  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_command({"prefix": "osd crush set-device-class", "class": "hdd",
 "ids": ["0"]} v 0) v1 -- 0x55bb4602de00 con 0x55bb45559200
2020-07-27 01:36:46.191 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 5 ==== mon_command_ack([{"prefix": "osd crush set-device-class", "class": "hdd", "ids": [
"0"]}]=0 osd.0 already set to class hdd. set-device-class item id 0 name 'osd.0' device_class 'hdd': no change
.  v1713) v1 ==== 207+0+0 (crc 0 0 0) 0x55bb46048480 con 0x55bb45559200
```

```
2020-07-27 01:36:46.191 7f02e472fc00  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_command({"prefix": "osd crush create-or-move", "id": 0, "weight"
:0.0000, "args": ["host=rnc03c-5fjd", "root=default"]} v 0) v1 -- 0x55bb4602e000 con 0x55bb45559200
2020-07-27 01:36:46.195 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 6 ==== mon_command_ack([{"prefix": "osd crush create-or-move", "id": 0, "weight":0.0000,
"args": ["host=rnc03c-5fjd", "root=default"]}]=0 create-or-move updated item name 'osd.0' weight 0 at location
 {host=rnc03c-5fjd,root=default} to crush map v1713) v1 ==== 250+0+0 (crc 0 0 0) 0x55bb448961c0 con 0x55bb4555
9200
2020-07-27 01:36:46.195 7f02e472fc00  0 osd.0 1396 done with init, starting boot process
2020-07-27 01:36:46.195 7f02e472fc00  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_subscribe({mgrmap=0+,osd_pg_creates=0+}) v3 -- 0x55bb4602f600 co
n 0x55bb45559200
2020-07-27 01:36:46.195 7f02e472fc00  1 osd.0 1396 start_boot
2020-07-27 01:36:46.195 7f02e472fc00  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_get_version(what=osdmap handle=1) v1 -- 0x55bb460381e0 con 0x55b
b45559200
2020-07-27 01:36:46.195 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 7 ==== mgrmap(e 45) v1 ==== 10944+0+0 (crc 0 0 0) 0x55bb46064000 con 0x55bb45559200
2020-07-27 01:36:46.195 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 8 ==== mon_get_version_reply(handle=1 version=1713) v2 ==== 24+0+0 (crc 0 0 0) 0x55bb448f
3880 con 0x55bb45559200
2020-07-27 01:36:46.195 7f02dc4e0700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_subscribe({osdmap=1397}) v3 -- 0x55bb46060c00 con 0x55bb45559200
2020-07-27 01:36:46.199 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 9 ==== osd_map(1397..1401 src has 1011..1713) v4 ==== 2697+0+0 (crc 0 0 0) 0x55bb44923200
 con 0x55bb45559200
2020-07-27 01:36:46.199 7f02dc7e3700  0 log_channel(cluster) log [WRN] : failed to encode map e1397 with expec
ted crc
2020-07-27 01:36:46.199 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] --> [v2:10.
75.91.77:3300/0,v1:10.75.91.77:6789/0] -- mon_get_osdmap(full 1397-1401) v1 -- 0x55bb46039680 con 0x55bb455592
00
*** Caught signal (Segmentation fault) **
 in thread 7f02dcbe7700 thread_name:cfin
2020-07-27 01:36:46.203 7f02dc7e3700  1 -- [v2:10.75.3.43:6800/1694128,v1:10.75.3.43:6801/1694128] <== mon.1 v
2:10.75.91.77:3300/0 10 ==== osd_map(1397..1401 src has 1011..1713) v4 ==== 292093+0+0 (crc 0 0 0) 0x55bb44922
f80 con 0x55bb45559200
 ceph version 14.2.10 (b340acf629a010a74d90da5782a2c5fe0b54ac20) nautilus (stable)
```

#11 - 07/27/2020 10:27 AM - Xiaoxi Chen

- File log.tar.gz added

- Affected Versions v15.2.4 added

- Affected Versions deleted (v14.2.10)

The issue also persist in latest Octopus release.

#12 - 07/27/2020 10:28 AM - Xiaoxi Chen

- Affected Versions v14.2.10 added

#13 - 07/27/2020 11:57 AM - Dan van der Ster

Maybe this will fix (untested -- use on a test cluster first):

```
~/g/c/s/osd (master|✚1) $ git diff
diff --git a/src/osd/OSD.cc b/src/osd/OSD.cc
index 3856294f8e..0801fafcb5 100644
--- a/src/osd/OSD.cc
+++ b/src/osd/OSD.cc
@@ -7949,6 +7949,12 @@ void OSD::handle_osd_map(MOSDMap *m)
         delete o;
         request_full_map(e, last);
         last = e - 1;
+
+        // don't continue committing if we failed to enc the first inc map
+        if (last < first) {
+          m->put();
+          return;
+        }
         break;
       }
       got_full_map(e);
```

I'm checking how to add a make check test for this scenario then see if this fixes reliably.

#14 - 07/27/2020 03:31 PM - Dan van der Ster

- Regression changed from No to Yes

- Severity changed from 2 - major to 1 - critical

Ahh now I understand why v14.2.10 crashes: fa842716b6dc3b2077e296d388c646f1605568b0 changed the `osdmap` in _committed_osd_maps from

being the class member OSD::osdmap to a local variable initialized to null. So when we run OSD::_committed_osd_maps with last < first, then v14.2.10 will crash with the null dereference at osdmap->is_up.

We can fix either by my patch above (which is coincidentally the same as that found in #45670), and/or we could do:

```
diff --git a/src/osd/OSD.cc b/src/osd/OSD.cc
index 8d9ab9caa5..4fe754ce55 100644
--- a/src/osd/OSD.cc
+++ b/src/osd/OSD.cc
@@ -8553,7 +8553,9 @@ void OSD::_committed_osd_maps(epoch_t first, epoch_t last, MOSDMap *m)
   bool do_shutdown = false;
   bool do_restart = false;
   bool network_error = false;
-  OSDMapRef osdmap;
+  OSDMapRef osdmap = get_osdmap();
+
+  ceph_assert(first <= last);

   // advance through the new maps
   for (epoch_t cur = first; cur <= last; cur++) {
```

Working on this here: https://github.com/ceph/ceph/pull/36297

#15 - 07/27/2020 04:00 PM - Dan van der Ster

- Assignee set to Dan van der Ster

- Backport set to nautilus, octopus

- Pull request ID set to 36297

- Component(RADOS) OSD added

#16 - 07/27/2020 04:02 PM - Neha Ojha

- Status changed from New to Fix Under Review

- Priority changed from Normal to Urgent

#17 - 07/27/2020 04:18 PM - Xiaoxi Chen

- Affected Versions v15.2.1, v15.2.2, v15.2.3 added

update affected version as it impacted all octopus release

#18 - 07/27/2020 04:19 PM - Xiaoxi Chen

- Subject changed from v14.2.10 Nautilus ceph_osd crash in _committed_osd_maps to ceph_osd crash in _committed_osd_maps when failed to encode first inc map

#19 - 07/27/2020 05:45 PM - Nathan Cutler

- Related to Bug #43903: osd segv in ceph::buffer::v14_2_0::ptr::release (PGTempMap::decode) added

#20 - 07/29/2020 01:11 AM - Neha Ojha

- Status changed from Fix Under Review to Pending Backport

#21 - 07/29/2020 04:29 AM - Nathan Cutler

- Copied to Backport #46741: nautilus: ceph_osd crash in _committed_osd_maps when failed to encode first inc map added

#22 - 07/29/2020 04:30 AM - Nathan Cutler

- Copied to Backport #46742: octopus: ceph_osd crash in _committed_osd_maps when failed to encode first inc map added

#23 - 08/13/2020 08:47 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved

While running with --resolve-parent, the script "backport-create-issue" noticed that all backports of this issue are in status "Resolved" or "Rejected".

Files

crash.a6cf1fdd-5cd6-4355-86d3-bbd39a4d8164.tar.gz	136 KB	07/10/2020	Markus Binz
log.tar.gz	362 KB	07/27/2020	Xiaoxi Chen