bluestore - Bug #50017

OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag

03/26/2021 04:36 PM - Jonas Jelten

Status:	Resolved	**% Done:**	0%
Priority:	High		
Assignee:			
Category:			
Target version:			
Source:		**Affected Versions:**	v15.2.10
Tags:		**ceph-qa-suite:**	
Backport:	pacific, octopus, nautilus	**Pull request ID:**	41429
Regression:	Yes	**Crash signature (v1):**	
Severity:	2 - major	**Crash signature (v2):**	
Reviewed:			

Description

I just started upgrading a cluster to octopus. MON and MGR are all on 15.2.10 and everything looks nice.

Then I upgraded the OSDs of one host: ceph-osd:amd64 (14.2.16-1bionic, 15.2.10-1bionic).
It found a lot of zombie spanning blobs, and then crashed. Now it refuses to start. And all the other OSDs on that host, too.

```
[... many more zombie blobs ...]
   -33> 2021-03-26T16:58:06.263+0100 7f284e679700 -1 bluestore(/var/lib/ceph/osd/ceph-80) fsck err
or: 6#31:210b8e42:datensicherungsserver::rbd_data.30.c2dfccc083f516.0000000000006852:head# - 1 zom
bie spanning blob(s) found, the first one: Blob(0x55d1c19c1100 spanning 0 blob([!~30000] csum crc3
2c/0x1000) use_tracker(0x3*0x10000 0x[0,0,0]) SharedBlob(0x55d1c19bf420 sbid 0x0))
   -32> 2021-03-26T16:58:06.675+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2631933952 unmapped: 750092288 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -31> 2021-03-26T16:58:06.711+0100 7f2860674d80 -1 bluestore(/var/lib/ceph/osd/ceph-80) fsck err
or: 6#31:2aeb130f:datensicherungsserver::rbd_data.30.c2dfccc083f516.0000000000006016:head# - 1 zom
bie spanning blob(s) found, the first one: Blob(0x55d162b94bc0 spanning 0 blob([!~10000] csum crc3
2c/0x1000) use_tracker(0x10000 0x0) SharedBlob(0x55d162b952c0 sbid 0x0))
   -30> 2021-03-26T16:58:06.967+0100 7f2860674d80  0 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_ch
eck_objects partial offload, done myself 572127 of 4822744objects, threads 2
   -29> 2021-03-26T16:58:06.983+0100 7f2860674d80  1 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open checking shared_blobs
   -28> 2021-03-26T16:58:07.675+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2612781056 unmapped: 769245184 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -27> 2021-03-26T16:58:08.179+0100 7f284f67b700  5 bluestore.MempoolThread(0x55d136754a98) _resi
ze_shards cache_size: 134217728 kv_alloc: 67108864 kv_used: 66050416 meta_alloc: 67108864 meta_use
d: 376573864 data_alloc: 67108864 data_used: 0
   -26> 2021-03-26T16:58:08.679+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2605285376 unmapped: 776740864 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -25> 2021-03-26T16:58:09.679+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2598166528 unmapped: 783859712 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -24> 2021-03-26T16:58:10.683+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2596691968 unmapped: 785334272 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -23> 2021-03-26T16:58:11.683+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2602442752 unmapped: 779583488 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -22> 2021-03-26T16:58:12.682+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2603786240 unmapped: 778240000 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -21> 2021-03-26T16:58:13.186+0100 7f284f67b700  5 bluestore.MempoolThread(0x55d136754a98) _resi
ze_shards cache_size: 134217728 kv_alloc: 67108864 kv_used: 66978032 meta_alloc: 67108864 meta_use
d: 320983536 data_alloc: 67108864 data_used: 0
   -20> 2021-03-26T16:58:13.686+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2603868160 unmapped: 778158080 heap: 3382026240 old mem: 134217728 new mem: 134217728
   -19> 2021-03-26T16:58:14.686+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 2496552960 unmapped: 885473280 heap: 3382026240 old mem: 134217728 new mem: 134217728
```

```
   -18> 2021-03-26T16:58:15.686+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 1615413248 unmapped: 1766612992 heap: 3382026240 old mem: 134217728 new mem: 353696767
   -17> 2021-03-26T16:58:15.702+0100 7f2860674d80  1 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open checking pool_statfs
   -16> 2021-03-26T16:58:15.702+0100 7f2860674d80  5 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open marking per_pool_omap=1
   -15> 2021-03-26T16:58:15.702+0100 7f2860674d80  5 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open applying repair results
   -14> 2021-03-26T16:58:16.070+0100 7f2860674d80 -1 rocksdb: submit_common error: Corruption: unk
nown WriteBatch tag code = 2 Rocksdb transaction:
Put( Prefix = O key = 0x80800000000000000a19400f'>dumpsite!rbd_data.6.28423ad8f48ca1.0000000002303
7b7!='0xfffffffffffffffeffffffffffffffff'o' Value size = 1388)
Put( Prefix = O key = 0x80800000000000000a194023'8dumpsite!rbd_data.6.28423ad8f48ca1.0000000002947
87d!='0xfffffffffffffffeffffffffffffffff'o' Value size = 1593)
Put( Prefix = O key = 0x80800000000000000a194027'Rdumpsite!rbd_data.6.28423ad8f48ca1.0000000001b36
6ff!='0xfffffffffffffffeffffffffffffffff'o' Value size = 2964)
Put( Prefix = O key = 0x80800000000000000a1940f69264756d'psite!rbd_data.6.28423ad8f48ca1.000000000
11bdd0c!='0xfffffffffffffffeffffffffffffffff'o' Value size = 2435)
Put( Prefix = O key = 0x80800000000000000a194109e164756d'psite!rbd_data.6.28423ad8f48ca1.000000000
1817e76!='0xfffffffffffffffeffffffffffffffff'o' Value size = 442)
Put( Prefix = O key = 0x80800000000000000a19411a9864756d'psite!rbd_data.6.28423ad8f48ca1.000000000
185cc48!='0xfffffffffffffffeffffffffffffffff'o' Value size = 4745)
[.........]
   -13> 2021-03-26T16:58:16.070+0100 7f2860674d80  5 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open repair applied
   -12> 2021-03-26T16:58:16.070+0100 7f2860674d80  2 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open 4822744 objects, 1280857 of them sharded.
   -11> 2021-03-26T16:58:16.070+0100 7f2860674d80  2 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open 27050327 extents to 25858545 blobs, 1119809 spanning, 1803008 shared.
   -10> 2021-03-26T16:58:16.070+0100 7f2860674d80  1 bluestore(/var/lib/ceph/osd/ceph-80) _fsck_on
_open <<<FINISH>>> with 14849 errors, 2 warnings, 14851 repaired, 0 remaining in 460.305791 second
s
    -9> 2021-03-26T16:58:16.690+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 1263484928 unmapped: 2118541312 heap: 3382026240 old mem: 353696767 new mem: 627999013
    -8> 2021-03-26T16:58:16.786+0100 7f2860674d80  2 osd.80 0 journal looks like hdd
    -7> 2021-03-26T16:58:16.786+0100 7f2860674d80  2 osd.80 0 boot
    -6> 2021-03-26T16:58:16.874+0100 7f2860674d80  1 osd.80 489295 init upgrade snap_mapper (first
 start as octopus)
    -5> 2021-03-26T16:58:17.690+0100 7f284f67b700  5 prioritycache tune_memory target: 2147483648
mapped: 1118093312 unmapped: 2263932928 heap: 3382026240 old mem: 627999013 new mem: 815929835
    -4> 2021-03-26T16:58:18.190+0100 7f284f67b700  5 bluestore.MempoolThread(0x55d136754a98) _resi
ze_shards cache_size: 815929835 kv_alloc: 339738624 kv_used: 66977632 meta_alloc: 297795584 meta_u
sed: 22052828 data_alloc: 171966464 data_used: 0
    -3> 2021-03-26T16:58:18.450+0100 7f284ae72700  5 bluestore(/var/lib/ceph/osd/ceph-80) _kv_sync
_thread utilization: idle 462.729599522s of 462.807117764s, submitted: 0
    -2> 2021-03-26T16:58:18.450+0100 7f284ae72700 -1 rocksdb: submit_common error: Corruption: unk
nown WriteBatch tag code = 2 Rocksdb transaction:
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_14_00000000000000BB_' Value size = 8
4)
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_14_00000000000000BD_' Value size = 9
1)
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_14_00000000000000C0_' Value size = 8
4)
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_14_00000000000000C2_' Value size = 9
1)
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_14_00000000000000C5_' Value size = 8
4)
[......]
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_9_00000000000000E7_' Value size = 73
)
Put( Prefix = m key = 0x0000000000000000000000000000402'.SNA_9_00000000000000EC_' Value size = 73
)
    -1> 2021-03-26T16:58:18.454+0100 7f284ae72700 -1 /build/ceph-15.2.10/src/os/bluestore/BlueStor
e.cc: In function 'void BlueStore::_txc_apply_kv(BlueStore::TransContext*, bool)' thread 7f284ae72
700 time 2021-03-26T16:58:18.457201+0100
/build/ceph-15.2.10/src/os/bluestore/BlueStore.cc: 11849: FAILED ceph_assert(r == 0)
```

```
 ceph version 15.2.10 (27917a557cca91e4da407489bbaa64ad4352cc02) octopus (stable)
 1: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x154) [0x55d12c437c02]
 2: (ceph::__ceph_assertf_fail(char const*, char const*, int, char const*, char const*, ...)+0) [0
x55d12c437ddd]
 3: (BlueStore::_txc_apply_kv(BlueStore::TransContext*, bool)+0x3a0) [0x55d12c970c00]
 4: (BlueStore::_kv_sync_thread()+0x12d3) [0x55d12c9ca3f3]
 5: (BlueStore::KVSyncThread::entry()+0xd) [0x55d12c9ef54d]
 6: (()+0x76db) [0x7f285e7ca6db]
 7: (clone()+0x3f) [0x7f285d56a71f]

[ OSD REBOOT by systemd ]

2021-03-26T16:58:29.198+0100 7fb6154e7d80  0 set uid:gid to 64045:64045 (ceph:ceph)
2021-03-26T16:58:29.198+0100 7fb6154e7d80  0 ceph version 15.2.10 (27917a557cca91e4da407489bbaa64a
d4352cc02) octopus (stable), process ceph-osd, pid 2044931
2021-03-26T16:58:29.198+0100 7fb6154e7d80  0 pidfile_write: ignore empty --pid-file
2021-03-26T16:58:29.198+0100 7fb6154e7d80  1 bdev create path /var/lib/ceph/osd/ceph-80/block type
 kernel
2021-03-26T16:58:29.198+0100 7fb6154e7d80  1 bdev(0x557a59578000 /var/lib/ceph/osd/ceph-80/block)
open path /var/lib/ceph/osd/ceph-80/block
2021-03-26T16:58:29.198+0100 7fb6154e7d80  1 bdev(0x557a59578000 /var/lib/ceph/osd/ceph-80/block)
open size 8001524072448 (0x74700000000, 7.3 TiB) block_size 4096 (4 KiB) rotational discard not su
pported
2021-03-26T16:58:29.198+0100 7fb6154e7d80  1 bluestore(/var/lib/ceph/osd/ceph-80) _set_cache_sizes
 cache_size 1073741824 meta 0.4 kv 0.4 data 0.2
[...]
2021-03-26T16:58:30.070+0100 7fb6154e7d80  1 bluefs add_block_device bdev 1 path /var/lib/ceph/osd
/ceph-80/block size 7.3 TiB
2021-03-26T16:58:30.070+0100 7fb6154e7d80  1 bluefs mount
2021-03-26T16:58:30.070+0100 7fb6154e7d80  1 bluefs _init_alloc id 1 alloc_size 0x10000 size 0x747
00000000
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option compaction_readahead_size = 20971
52
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option compression = kNoCompression
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option max_background_compactions = 2
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option max_write_buffer_number = 4
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option min_write_buffer_number_to_merge
= 1
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option recycle_log_file_num = 4
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option writable_file_max_buffer_size = 0
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option write_buffer_size = 268435456
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option compaction_readahead_size = 20971
52
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option compression = kNoCompression
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option max_background_compactions = 2
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option max_write_buffer_number = 4
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option min_write_buffer_number_to_merge
= 1
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option recycle_log_file_num = 4
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option writable_file_max_buffer_size = 0
2021-03-26T16:58:30.238+0100 7fb6154e7d80  0  set rocksdb option write_buffer_size = 268435456
2021-03-26T16:58:42.438+0100 7fb6154e7d80 -1 rocksdb: Corruption: unknown WriteBatch tag
2021-03-26T16:58:42.438+0100 7fb6154e7d80 -1 bluestore(/var/lib/ceph/osd/ceph-80) _open_db errorin
g opening db:
2021-03-26T16:58:42.438+0100 7fb6154e7d80  1 bluefs umount
2021-03-26T16:58:42.438+0100 7fb6154e7d80  1 fbmap_alloc 0x557a58888900 shutdown
2021-03-26T16:58:42.462+0100 7fb6154e7d80  1 bdev(0x557a59578380 /var/lib/ceph/osd/ceph-80/block)
close
2021-03-26T16:58:42.642+0100 7fb6154e7d80  1 bdev(0x557a59578000 /var/lib/ceph/osd/ceph-80/block)
close
2021-03-26T16:58:42.910+0100 7fb6154e7d80 -1 osd.80 0 OSD:init: unable to mount object store
2021-03-26T16:58:42.910+0100 7fb6154e7d80 -1  ** ERROR: osd init failed: (5) Input/output error
```

And that's it, it now refuses to start.

Full log is attached.

What do?

Related issues:

Related to bluestore - Bug #48216: Spanning blobs list might have zombie blob...	**New**
Copied to bluestore - Backport #50938: pacific: OSDs broken after nautilus->o...	**Resolved**
Copied to bluestore - Backport #50939: nautilus: OSDs broken after nautilus->...	**Resolved**
Copied to bluestore - Backport #50940: octopus: OSDs broken after nautilus->o...	**Resolved**

History

#1 - 03/26/2021 05:31 PM - Jonas Jelten

- Subject changed from OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag code = 2 to OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag

All OSDs on that host except two are now corrupted.
These two hang during fsck with 100% load. One OSDs has one ShallowFSCK thread at 100%, the other OSD has 2 100% ShallowFSCK threads.

For one of the threads, perf top says this:

```
  Children      Self  Shared Object          Symbol
+   63.90%    63.85%  ceph-osd               [.] ShallowFSCKThreadPool::FSCKWorkQueue<256ul>::_void_dequeue
+   33.33%    33.32%  libtcmalloc.so.4.3.0   [.] tcmalloc::PageHeap::AllocLarge
+    7.44%     0.00%  [unknown]              [.] 0x8d4818ec83485355
+    7.44%     0.00%  ceph-osd               [.] ShallowFSCKThreadPool::FSCKWorkQueue<256ul>::~FSCKWorkQueue
+    7.44%     0.00%  [unknown]              [.] 0x00005623866f4798
+    3.78%     0.00%  [unknown]              [.] 0x000000000000489b
+    1.74%     1.74%  libc-2.27.so           [.] random
     0.43%     0.43%  libc-2.27.so           [.] random_r
     0.27%     0.27%  ceph-osd               [.] ShallowFSCKThreadPool::worker
```

#2 - 04/08/2021 05:53 PM - Jonas Jelten

Another attempt on a different host, now we upgrade just one 1T device...
I've set ceph config set osd bluestore_fsck_quick_fix_threads 1 since a race between the two ShallowFSCK threads seems likely.

But the upgrade seems to be failing again, the OSD doesn't report any more zombie spanning blobs, but consumes 100% cpu, in the main thread. ShallowFSCK thread has 0%. The log is silent, it doesn't react to daemon commands.

Attaching GDB:

```
#0  0x00007fed1ce91620 in tcmalloc::PageHeap::AllocLarge(unsigned long) () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#1  0x00007fed1ce91e4f in tcmalloc::PageHeap::New(unsigned long) () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#2  0x00007fed1ce907bf in tcmalloc::CentralFreeList::Populate() () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#3  0x00007fed1ce909c8 in tcmalloc::CentralFreeList::FetchFromOneSpansSafe(int, void**, void**) () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#4  0x00007fed1ce90abf in tcmalloc::CentralFreeList::RemoveRange(void**, void**, int) () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#5  0x00007fed1ce93a5a in tcmalloc::ThreadCache::FetchFromCentralCache(unsigned long, unsigned long) () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#6  0x00007fed1cea495b in tc_newarray () from target:/usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#7  0x0000561a1c78adcb in mempool::pool_allocator<(mempool::pool_index_t)8, BlueStore::Blob>::allocate (this=0x561a1d986e40 <mempool::bluestore_Blob::alloc_bluestore_blob>, n=1, p=0x0) at ./src/include/mempool.h:333
#8  BlueStore::Blob::operator new (size=<optimized out>) at ./src/os/bluestore/BlueStore.cc:72
#9  0x0000561a1c7ab8ba in BlueStore::ExtentMap::decode_some (this=this@entry=0x561a5dd2f550, bl=...) at ./src/os/bluestore/BlueStore.cc:3138
#10 0x0000561a1c7acd0c in BlueStore::Onode::decode (c=..., oid=..., key=..., v=...) at ./src/os/bluestore/BlueStore.cc:3620
#11 0x0000561a1c807c2b in BlueStore::fsck_check_objects_shallow (this=this@entry=0x561a27544000, depth=BlueStore::FSCK_SHALLOW, pool_id=<optimized out>, c=..., oid=..., key=..., value=..., expecting_shards=0x7fff49b7d9e0, referenced=0x7fff49b7d880, ctx=...)
    at ./src/os/bluestore/BlueStore.cc:7683
#12 0x0000561a1c80bb80 in BlueStore::_fsck_check_objects (this=this@entry=0x561a27544000, depth=depth@entry=BlueStore::FSCK_SHALLOW, ctx=...) at ./src/os/bluestore/BlueStore.cc:8339
#13 0x0000561a1c80fd4a in BlueStore::_fsck_on_open (this=this@entry=0x561a27544000, depth=depth@entry=BlueStore::FSCK_SHALLOW, repair=repair@entry=true) at ./src/os/bluestore/BlueStore.cc:8711
#14 0x0000561a1c82b9cd in BlueStore::_mount (this=0x561a27544000, kv_only=<optimized out>, open_db=<optimized out>) at ./src/os/bluestore/BlueStore.cc:7359
#15 0x0000561a1c33a381 in OSD::init (this=0x561a274ec000) at ./src/osd/OSD.cc:3291
#16 0x0000561a1c2a52cc in main (argc=<optimized out>, argv=<optimized out>) at ./src/ceph_osd.cc:711
```

perf top says:

```
  Children      Self  Shared Object         Symbol

-   99.92%    98.21%  libtcmalloc.so.4.3.0  [.] tcmalloc::PageHeap::AllocLarge

   - 98.21% 0x19f

        tcmalloc::PageHeap::AllocLarge

-   10.96%     0.00%  [unknown]             [.] 0x000000000000019f

     0x19f

     tcmalloc::PageHeap::AllocLarge

-    0.85%     0.00%  [kernel]              [k] irq_exit

-    0.84%     0.01%  [kernel]              [k] __softirqentry_text_start
```

I've restarted the OSD, and it ran through the analysis this time, but corrupted again when committing.

```
[many more zombie blobs]
   -24> 2021-04-08T19:48:19.323+0200 7f0238a2ed80 -1 bluestore(/var/lib/ceph/osd/ceph-90) fsck error: 6#10:e55
ce554:dumpsite::rbd_data.6.28423ad8f48ca1.00000000017e7410:head# - 1 zombie spanning blob(s) found, the first
one: Blob(0x559eb277ce60 spanning 2 blob([!~30000] csum crc32c/0
x1000) use_tracker(0x3*0x10000 0x[0,0,0]) SharedBlob(0x559eb277c530 sbid 0x0))
   -23> 2021-04-08T19:48:19.323+0200 7f0238a2ed80  0 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_check_objects
partial offload, done myself 134108 of 1017671objects, threads 1
   -22> 2021-04-08T19:48:19.331+0200 7f0238a2ed80  1 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open checki
ng shared_blobs
   -21> 2021-04-08T19:48:19.387+0200 7f0227a37700  5 prioritycache tune_memory target: 2147483648 mapped: 1714
405376 unmapped: 5177344 heap: 1719582720 old mem: 1020054729 new mem: 1020054729
   -20> 2021-04-08T19:48:20.391+0200 7f0227a37700  5 prioritycache tune_memory target: 2147483648 mapped: 1712
783360 unmapped: 6799360 heap: 1719582720 old mem: 1020054729 new mem: 1020054729
   -19> 2021-04-08T19:48:21.215+0200 7f0238a2ed80  1 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open checki
ng pool_statfs
   -18> 2021-04-08T19:48:21.215+0200 7f0238a2ed80  5 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open markin
g per_pool_omap=1
   -17> 2021-04-08T19:48:21.215+0200 7f0238a2ed80  5 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open applyi
ng repair results
   -16> 2021-04-08T19:48:21.359+0200 7f0238a2ed80 -1 rocksdb: submit_common error: Corruption: unknown WriteBa
tch tag code = 2 Rocksdb transaction:
Put( Prefix = O key = 0x80800000000000000a20a017eb64756d'psite!rbd_data.6.28423ad8f48ca1.000000000153d8bd!='0x
fffffffffffffffeffffffffffffffff'o' Value size = 442)
Put( Prefix = O key = 0x80800000000000000a20a023e264756d'psite!rbd_data.6.28423ad8f48ca1.000000000133fc02!='0x
fffffffffffffffeffffffffffffffff'o' Value size = 2622)
Put( Prefix = O key = 0x80800000000000000a20a02e'?dumpsite!rbd_data.6.28423ad8f48ca1.00000000019e2a6d!='0xffff
ffffffffffeffffffffffffffff'o' Value size = 2198)
[... more transaction Put values ...]
   -15> 2021-04-08T19:48:21.359+0200 7f0238a2ed80  5 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open repair
 applied
   -14> 2021-04-08T19:48:21.359+0200 7f0238a2ed80  2 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open 101767
1 objects, 233150 of them sharded.
   -13> 2021-04-08T19:48:21.359+0200 7f0238a2ed80  2 bluestore(/var/lib/ceph/osd/ceph-90) _fsck_on_open 552514
5 extents to 5275685 blobs, 257172 spanning, 322925 shared.
[... boot continues and crashes as in the top post ...]
```

So it's not a threading issue, ok.

I have plenty more OSDs to try, lol.

btw the "repair applied" success result is obviously wrong, I've already fixed that in https://github.com/ceph/ceph/pull/40444

#3 - 05/05/2021 05:38 PM - Jonas Jelten

Ok, some new information, tested on 15.2.11 :D

It seems that the OSDs are shredded with the ceph-osd boot-time fsck, but not with the ceph-bluestore-tool.

```
sudo ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-121 --command repair
```

So maybe there's some subtle difference between the bluestoretool and ceph-osd? The most obvious difference is repair vs quickfix. So let's try to run:

```
sudo ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-92 --command quick-fix
```

This hangs in the main thread at 100% load (two ShallowFSCK threads at 0%).

```
#0  0x00007ffff7b85620 in tcmalloc::PageHeap::AllocLarge(unsigned long) () from /usr/lib/x86_64-linux-gnu/libt
cmalloc.so.4
#1  0x00007ffff7b85e4f in tcmalloc::PageHeap::New(unsigned long) () from /usr/lib/x86_64-linux-gnu/libtcmalloc
.so.4
#2  0x00007ffff7b847bf in tcmalloc::CentralFreeList::Populate() () from /usr/lib/x86_64-linux-gnu/libtcmalloc.
so.4
#3  0x00007ffff7b849c8 in tcmalloc::CentralFreeList::FetchFromOneSpansSafe(int, void**, void**) ()   from /usr
/lib/x86_64-linux-gnu/libtcmalloc.so.4
#4  0x00007ffff7b84abf in tcmalloc::CentralFreeList::RemoveRange(void**, void**, int) () from /usr/lib/x86_64-
linux-gnu/libtcmalloc.so.4
#5  0x00007ffff7b87a5a in tcmalloc::ThreadCache::FetchFromCentralCache(unsigned long, unsigned long) ()   from
 /usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#6  0x00007ffff7b9895b in tc_newarray () from /usr/lib/x86_64-linux-gnu/libtcmalloc.so.4
#7  0x0000555556149e18 in rocksdb::AllocateBlock(unsigned long, rocksdb::MemoryAllocator*) ()
#8  0x000055555614afd2 in rocksdb::BlockFetcher::PrepareBufferForBlockFromFile() ()
#9  0x000055555614869a in rocksdb::BlockFetcher::ReadBlockContents() ()
#10 0x000055555612f230 in rocksdb::BlockBasedTable::MaybeReadBlockAndLoadToCache(rocksdb::FilePrefetchBuffer*,
 rocksdb::BlockBasedTable::Rep*, rocksdb::ReadOptions const&, rocksdb::BlockHandle const&, rocksdb::Uncompress
ionDict const&, rocksdb::BlockBasedTable::CachableEntry<rocksdb::Block>*, bool, rocksdb::GetContext*) ()
#11 0x000055555613e04f in rocksdb::DataBlockIter* rocksdb::BlockBasedTable::NewDataBlockIterator<rocksdb::Data
BlockIter>(rocksdb::BlockBasedTable::Rep*, rocksdb::ReadOptions const&, rocksdb::BlockHandle const&, rocksdb::
DataBlockIter*, bool, bool, bool, rocksdb::GetContext*, rocksdb::Status, rocksdb::FilePrefetchBuffer*) ()
#12 0x0000555556146cde in rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::InitDataBl
ock() ()
#13 0x0000555556146e85 in rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::FindKeyFor
ward() ()
#14 0x000055555614594e in rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::Next() ()
#15 0x000055555607efd8 in rocksdb::IteratorWrapperBase<rocksdb::Slice>::Next() ()
#16 0x000055555605d231 in rocksdb::(anonymous namespace)::LevelIterator::Next() ()
#17 0x000055555607efd8 in rocksdb::IteratorWrapperBase<rocksdb::Slice>::Next() ()
#18 0x0000555556157d23 in rocksdb::MergingIterator::Next() ()
#19 0x0000555555fc090a in rocksdb::DBIter::Next() ()
#20 0x0000555555fc9e7e in rocksdb::ArenaWrappedDBIter::Next() ()
#21 0x0000555555ec852e in RocksDBStore::RocksDBWholeSpaceIteratorImpl::next() () at ./src/kv/RocksDBStore.cc:1
377
#22 0x0000555555cf2025 in KeyValueDB::PrefixIteratorImpl::next (this=<optimized out>) at /usr/include/c++/9/bi
ts/shared_ptr_base.h:1020
#23 BlueStore::_fsck_check_objects(BlueStore::FSCKDepth, BlueStore::FSCK_ObjectCtx&) () at ./src/os/bluestore/
BlueStore.cc:8237
#24 0x0000555555cf6924 in BlueStore::_fsck_on_open(BlueStore::FSCKDepth, bool) () at ./src/os/bluestore/BlueSt
ore.cc:8711
#25 0x0000555555d02326 in BlueStore::_fsck(BlueStore::FSCKDepth, bool) () at ./src/os/bluestore/BlueStore.cc:8
546
#26 0x0000555555c03f12 in BlueStore::quick_fix (this=0x7fffffffadc0) at ./src/os/bluestore/BlueStore.h:2526
#27 main () at ./src/os/bluestore/bluestore_tool.cc:432
#28 0x00007fffece2abf7 in __libc_start_main (main=0x555555c02a60 <main>, argc=5, argv=0x7fffffffe988, init=<op
timized out>,    fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffffffe978) at ../csu/libc-sta
rt.c:310
#29 0x0000555555c0b29a in _start () at /usr/include/c++/9/bits/char_traits.h:335
```

Tested several times, it hangs. After killing this, the bluestoretool repair did succeed though, and the OSD was de-zombie-blobbed, and works like it

should.

So: It seems to be quick-fix vs repair? At least with bluestoretool-quickfix it didn't shred, so far.

#4 - 05/07/2021 08:04 PM - Dan van der Ster

FTR, Igor replied on the ML:

> I think the root cause is related to the high amount of repairs made during the first post-upgrade fsck run.
>
> The check (and fix) for zombie spanning blobs was been backported to v15.2.9 (here is the PR https://github.com/ceph/ceph/pull/39256). And I presumt it's the one which causes BlueFS data corruption due to huge transaction happening during such a repair.
>
> I haven't seen this exact issue (as having that many zombie blobs is a rarely met bug by itself) but we had to some degree similar issue with upgrading omap names, see: https://github.com/ceph/ceph/pull/39377
>
> Huge resulting transaction could cause too big write to WAL which in turn caused data corruption (see https://github.com/ceph/ceph/pull/39701)
>
> Although the fix for the latter has been merged for 15.2.10 some additional issues with huge transactions might still exist...
>
> If someone can afford another OSD loss it could be interesting to get an OSD log for such a repair with debug-bluefs set to 20...
>
> I'm planning to make a fix to cap transaction size for repair in the nearest future anyway though..

Igor is there a tracker for that "cap transaction size for repair" ?

Jonas could you provide a log of repair with `debug_bluefs = 20` ?

#5 - 05/08/2021 06:34 PM - Konstantin Shalygin

Jonas, looks like my https://tracker.ceph.com/issues/48216#note-3 ?
This cluster have a EC meta pool for RBD?
Also, the repair:

```
sudo ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-121 --command repair
```

Before run new version of ceph-osd will fix this OSD's for you?

#6 - 05/11/2021 08:28 AM - Konstantin Shalygin

Checked on next host by myself - "--command repair" fix OSD's before Nautilus auto fsck, and also CAN repair already broken OSD's (but not always)

#7 - 05/13/2021 11:13 PM - Igor Fedotov

- Related to Bug #48216: Spanning blobs list might have zombie blobs that aren't of use any more added

#8 - 05/14/2021 05:19 PM - Jonas Jelten

Yes, this de-zombie-blobs the OSDs. So now I have an upgradepath by (automatically) stopping an osd, running bluestoretool-repair, and then starting it with ceph 15. To be sure I disabled the quickfsck-on-boot for now.

bluestoretool couldn't repair the broken OSDs, the rocksdb won't open since its journal is corrupted.

Konstantin: I had the zombie spanning-blobs for ec data pools used by cephfs and rbd.

Igor: Do you know why the boot-quickfix fails, but the bluestoretool repair works (as in not corrupting rocksdb)? Shouldn't that be the same mechanism?

I'd like to avoid losing more OSDs at the moment, but if you're gonna invest time, I can run a debug-bluefs-20 quickfix-crash again (wondering though why bluefs20 would provide any hint since it seems to be uninvolved, rather the fault is in repair+rocksdb?).

#9 - 05/14/2021 11:28 PM - Igor Fedotov

Jonas Jelten wrote:

> Yes, this de-zombie-blobs the OSDs. So now I have an upgradepath by (automatically) stopping an osd, running bluestoretool-repair, and then starting it with ceph 15. To be sure I disabled the quickfsck-on-boot for now.
>
> bluestoretool couldn't repair the broken OSDs, the rocksdb won't open since its journal is corrupted.
>
> Konstantin: I had the zombie spanning-blobs for ec data pools used by cephfs and rbd.
>
> Igor: Do you know why the boot-quickfix fails, but the bluestoretool repair works (as in not corrupting rocksdb)? Shouldn't that be the same mechanism?

One notable difference between repair and quick-fix is that the latter is multithreaded. By default 'bluestore_fsck_quick_fix_threads' parameter specifies two threads to perform checking. So one can verify the hypothesis by setting it to 1 and running a quick-fix.

Another difference is the narrower checking scope for quick-fix but I doubt this has that bad impact.

> I'd like to avoid losing more OSDs at the moment, but if you're gonna invest time, I can run a debug-bluefs-20 quickfix-crash again (wondering

> though why bluefs20 would provide any hint since it seems to be uninvolved, rather the fault is in repair+rocksdb?).

Perhaps we can postpone that for a bit - hopefully I'll have time to try to reproduce the issue in my lab next week. Looks like this should be doable... If this wouldn't succeeded then we can go back to the idea to dissect one of your OSDs.

#10 - 05/19/2021 11:25 PM - Igor Fedotov

- Status changed from New to Fix Under Review

- Backport set to pacific, octopus

- Pull request ID set to 41429

#11 - 05/20/2021 05:30 AM - Konstantin Shalygin

- Backport changed from pacific, octopus to pacific, octopus, nautilus

#12 - 05/20/2021 05:33 AM - Konstantin Shalygin

Added nautilus to backports, because if upgrade from luminous release, flow is luminous->nautilus->pacific.

#13 - 05/20/2021 10:21 AM - Igor Fedotov

Konstantin Shalygin wrote:

> Added nautilus to backports, because if upgrade from luminous release, flow is luminous->nautilus->pacific.

yep, missed that quick-fix is in Nautilus too..

#14 - 05/23/2021 12:42 AM - Kefu Chai

- Status changed from Fix Under Review to Pending Backport

#15 - 05/23/2021 12:46 AM - Backport Bot

- Copied to Backport #50938: pacific: OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag added

#16 - 05/23/2021 12:46 AM - Backport Bot

- Copied to Backport #50939: nautilus: OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag added

#17 - 05/23/2021 12:46 AM - Backport Bot

- Copied to Backport #50940: octopus: OSDs broken after nautilus->octopus upgrade: rocksdb Corruption: unknown WriteBatch tag added

#18 - 06/29/2021 08:11 AM - Loïc Dachary

- 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

convert-osd.80.log.xz	321 KB	03/26/2021	Jonas Jelten