bluestore - Bug #40741

Mass OSD failure, unable to restart

07/11/2019 07:26 PM - Brett Chancellor

Status:	Triaged	**% Done:**	0%
Priority:	Normal		
Assignee:	Igor Fedotov		
Category:			
Target version:	v14.2.1		
Source:		**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:		**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

Cluster: 14.2.1
OSDs: 250 spinners in default root, 63 SSDs in ssd root

History: 5 days ago, this cluster began losing spinning drives every couple of hours. Many of them were unable to be restarted once they went down, so they had to be rebuilt. After reaching out to the ceph users group we tried setting the bluestore_allocator and bluesfs_allocator to stupid. This allowed newly dying OSDs to be brought back online, although it didn't stop others from dying. Once the cluster finished rebalancing the performance was terrible (disks all idle and error free, individual clients getting 1-2 iops with 12k ms latency) with either allocator, We did note that performance was fine on any root other than the default root. OSDs continued to commit suicide every 30-40 minutes. In an attempt to improve performance we decided to try and move the <zone>.rgw.meta pool from spinning drives to SSD.

After doing this SSD's began to fail in mass. We are unable bring up the SSD OSDs with either the stupid or bitmap allocators.

Attached is an osd with debug_bluestore 10 set.

Current config
$ sudo ceph config dump

WHO	MASK	LEVEL	OPTION	VALUE	RO
global		advanced	bluestore_warn_on_bluefs_spillover	false	
global		advanced	mon_warn_pg_not_deep_scrubbed_ratio	0.000000	
global		advanced	mon_warn_pg_not_scrubbed_ratio	0.000000	
global		advanced	osd_deep_scrub_interval	1814400.000000	
global		advanced	osd_scrub_max_interval	1814400.000000	
global		advanced	osd_scrub_min_interval	259200.000000	
mon		advanced	mon_osd_down_out_interval	1200	
mon.ceph0rdi-mon1-1-prd		advanced	ms_bind_msgr2	false	
mon.ceph0rdi-mon2-1-prd		advanced	ms_bind_msgr2	false	
mon.ceph0rdi-mon3-1-prd		advanced	ms_bind_msgr2	false	
osd		advanced	bluestore_bluefs_gift_ratio	0.000200	
osd		advanced	osd_max_backfills	3	
osd		basic	osd_memory_target	4294967296	
osd		advanced	osd_op_thread_timeout	90	
osd		advanced	osd_recovery_sleep_hdd	0.000000	
osd		advanced	osd_recovery_sleep_hybrid	0.000000	

Example stack trace:
-1> 2019-07-11 18:59:24.296 7fcc1caf5d80 -1 /home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1/src/os/bluestore/BlueFS.cc: In function 'int BlueFS::_flush_range(BlueFS::FileWriter*, uint64_t, uint64_t)' thread 7fcc1caf5d80 time 2019-07-11 18:59:24.289089
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1/src/os/bluestore/BlueFS.cc: 2044: abort()

```
ceph version 14.2.1 (d555a9489eb35f84f2e1ef49b77e19da9d113972) nautilus (stable)
 1: (ceph::__ceph_abort(char const*, int, char const*, std::string const&)+0xd8) [0x560dbdbc9cd0]
 2: (BlueFS::_flush_range(BlueFS::FileWriter*, unsigned long, unsigned long)+0x1daf) [0x560dbe2819
af]
 3: (BlueFS::_flush(BlueFS::FileWriter*, bool)+0x10b) [0x560dbe281b4b]
 4: (BlueRocksWritableFile::Flush()+0x3d) [0x560dbe29f84d]
 5: (rocksdb::WritableFileWriter::Flush()+0x19e) [0x560dbe8cdd0e]
 6: (rocksdb::WritableFileWriter::Sync(bool)+0x2e) [0x560dbe8cdfee]
 7: (rocksdb::BuildTable(std::string const&, rocksdb::Env*, rocksdb::ImmutableCFOptions const&, ro
cksdb::MutableCFOptions const&, rocksdb::EnvOptions const&, rocksdb::TableCache*, rocksdb::Interna
lIteratorBase&lt;rocksdb::Slice&gt;*, std::unique_ptr&lt;rocksdb::InternalIteratorBase&lt;rocksdb:
:Slice&gt;, std::default_delete&lt;rocksdb::InternalIteratorBase&lt;rocksdb::Slice&gt; > >, rocksd
b::FileMetaData*, rocksdb::InternalKeyComparator const&, std::vector&lt;std::unique_ptr&lt;rocksdb
::IntTblPropCollectorFactory, std::default_delete&lt;rocksdb::IntTblPropCollectorFactory&gt; >, st
d::allocator&lt;std::unique_ptr&lt;rocksdb::IntTblPropCollectorFactory, std::default_delete&lt;roc
ksdb::IntTblPropCollectorFactory&gt; > > > const*, unsigned int, std::string const&, std::vector&l
t;unsigned long, std::allocator&lt;unsigned long&gt; >, unsigned long, rocksdb::SnapshotChecker*,
rocksdb::CompressionType, rocksdb::CompressionOptions const&, bool, rocksdb::InternalStats*, rocks
db::TableFileCreationReason, rocksdb::EventLogger*, int, rocksdb::Env::IOPriority, rocksdb::TableP
roperties*, int, unsigned long, unsigned long, rocksdb::Env::WriteLifeTimeHint)+0x2368) [0x560dbe8
fb978]
 8: (rocksdb::DBImpl::WriteLevel0TableForRecovery(int, rocksdb::ColumnFamilyData*, rocksdb::MemTab
le*, rocksdb::VersionEdit*)+0xc66) [0x560dbe7716a6]
 9: (rocksdb::DBImpl::RecoverLogFiles(std::vector&lt;unsigned long, std::allocator&lt;unsigned lon
g&gt; > const&, unsigned long*, bool)+0x1672) [0x560dbe7735a2]
 10: (rocksdb::DBImpl::Recover(std::vector&lt;rocksdb::ColumnFamilyDescriptor, std::allocator&lt;r
ocksdb::ColumnFamilyDescriptor&gt; > const&, bool, bool, bool)+0x809) [0x560dbe774b99]
 11: (rocksdb::DBImpl::Open(rocksdb::DBOptions const&, std::string const&, std::vector&lt;rocksdb:
:ColumnFamilyDescriptor, std::allocator&lt;rocksdb::ColumnFamilyDescriptor&gt; > const&, std::vect
or&lt;rocksdb::ColumnFamilyHandle*, std::allocator&lt;rocksdb::ColumnFamilyHandle*&gt; >
, rocksdb::DB*, bool, bool)+0x658) [0x560dbe7759a8]
 12: (rocksdb::DB::Open(rocksdb::DBOptions const&, std::string const&, std::vector&lt;rocksdb::Col
umnFamilyDescriptor, std::allocator&lt;rocksdb::ColumnFamilyDescriptor&gt; > const&, std::vector&l
t;rocksdb::ColumnFamilyHandle*, std::allocator&lt;rocksdb::ColumnFamilyHandle*&gt; >, rocksdb::DB
*)+0x24) [0x560dbe777184]
 13: (RocksDBStore::do_open(std::ostream&, bool, bool, std::vector&lt;KeyValueDB::ColumnFamily, st
d::allocator&lt;KeyValueDB::ColumnFamily&gt; > const*)+0x1660) [0x560dbe20bde0]
 14: (BlueStore::_open_db(bool, bool, bool)+0xf8e) [0x560dbe16077e]
 15: (BlueStore::_open_db_and_around(bool)+0x165) [0x560dbe17dcb5]
 16: (BlueStore::_mount(bool, bool)+0x6a4) [0x560dbe1ba694]
 17: (OSD::init()+0x3aa) [0x560dbdd30d7a]
 18: (main()+0x14fa) [0x560dbdbcd1da]
 19: (__libc_start_main()+0xf5) [0x7fcc185283d5]
 20: (()+0x564555) [0x560dbdcc1555]



0> 2019-07-11 18:59:24.304 7fcc1caf5d80 -1 *** Caught signal (Aborted) **
 in thread 7fcc1caf5d80 thread_name:ceph-osd



ceph version 14.2.1 (d555a9489eb35f84f2e1ef49b77e19da9d113972) nautilus (stable)
 1: (()+0xf5d0) [0x7fcc197455d0]
 2: (gsignal()+0x37) [0x7fcc1853c207]
 3: (abort()+0x148) [0x7fcc1853d8f8]
 4: (ceph::__ceph_abort(char const*, int, char const*, std::string const&)+0x19c) [0x560dbdbc9d94]
 5: (BlueFS::_flush_range(BlueFS::FileWriter*, unsigned long, unsigned long)+0x1daf) [0x560dbe2819
af]
 6: (BlueFS::_flush(BlueFS::FileWriter*, bool)+0x10b) [0x560dbe281b4b]
 7: (BlueRocksWritableFile::Flush()+0x3d) [0x560dbe29f84d]
 8: (rocksdb::WritableFileWriter::Flush()+0x19e) [0x560dbe8cdd0e]
 9: (rocksdb::WritableFileWriter::Sync(bool)+0x2e) [0x560dbe8cdfee]
 10: (rocksdb::BuildTable(std::string const&, rocksdb::Env*, rocksdb::ImmutableCFOptions const&, r
ocksdb::MutableCFOptions const&, rocksdb::EnvOptions const&, rocksdb::TableCache*, rocksdb::Intern
alIteratorBase&lt;rocksdb::Slice&gt;*, std::unique_ptr&lt;rocksdb::InternalIteratorBase&lt;rocksdb
::Slice&gt;, std::default_delete&lt;rocksdb::InternalIteratorBase&lt;rocksdb::Slice&gt; > >, rocks
```

```
db::FileMetaData*, rocksdb::InternalKeyComparator const&, std::vector&lt;std::unique_ptr&lt;rocksdb::IntTblPropCollectorFactory, std::default_delete&lt;rocksdb::IntTblPropCollectorFactory&gt; >, std::allocator&lt;std::unique_ptr&lt;rocksdb::IntTblPropCollectorFactory, std::default_delete&lt;rocksdb::IntTblPropCollectorFactory&gt; > > > const*, unsigned int, std::string const&, std::vector&lt;unsigned long, std::allocator&lt;unsigned long&gt; >, unsigned long, rocksdb::SnapshotChecker*, rocksdb::CompressionType, rocksdb::CompressionOptions const&, bool, rocksdb::InternalStats*, rocksdb::TableFileCreationReason, rocksdb::EventLogger*, int, rocksdb::Env::IOPriority, rocksdb::TableProperties*, int, unsigned long, unsigned long, rocksdb::Env::WriteLifeTimeHint)+0x2368) [0x560dbe8fb978]
 11: (rocksdb::DBImpl::WriteLevel0TableForRecovery(int, rocksdb::ColumnFamilyData*, rocksdb::MemTable*, rocksdb::VersionEdit*)+0xc66) [0x560dbe7716a6]
 12: (rocksdb::DBImpl::RecoverLogFiles(std::vector&lt;unsigned long, std::allocator&lt;unsigned long&gt; > const&, unsigned long*, bool)+0x1672) [0x560dbe7735a2]
 13: (rocksdb::DBImpl::Recover(std::vector&lt;rocksdb::ColumnFamilyDescriptor, std::allocator&lt;rocksdb::ColumnFamilyDescriptor&gt; > const&, bool, bool, bool)+0x809) [0x560dbe774b99]
 14: (rocksdb::DBImpl::Open(rocksdb::DBOptions const&, std::string const&, std::vector&lt;rocksdb::ColumnFamilyDescriptor, std::allocator&lt;rocksdb::ColumnFamilyDescriptor&gt; > const&, std::vector&lt;rocksdb::ColumnFamilyHandle*, std::allocator&lt;rocksdb::ColumnFamilyHandle*&gt; >
, rocksdb::DB*, bool, bool)+0x658) [0x560dbe7759a8]
 15: (rocksdb::DB::Open(rocksdb::DBOptions const&, std::string const&, std::vector&lt;rocksdb::ColumnFamilyDescriptor, std::allocator&lt;rocksdb::ColumnFamilyDescriptor&gt; > const&, std::vector&lt;rocksdb::ColumnFamilyHandle*, std::allocator&lt;rocksdb::ColumnFamilyHandle*&gt; >, rocksdb::DB*)+0x24) [0x560dbe777184]
 16: (RocksDBStore::do_open(std::ostream&, bool, bool, std::vector&lt;KeyValueDB::ColumnFamily, std::allocator&lt;KeyValueDB::ColumnFamily&gt; > const*)+0x1660) [0x560dbe20bde0]
 17: (BlueStore::_open_db(bool, bool, bool)+0xf8e) [0x560dbe16077e]
 18: (BlueStore::_open_db_and_around(bool)+0x165) [0x560dbe17dcb5]
 19: (BlueStore::_mount(bool, bool)+0x6a4) [0x560dbe1ba694]
 20: (OSD::init()+0x3aa) [0x560dbdd30d7a]
 21: (main()+0x14fa) [0x560dbdbcd1da]
 22: (__libc_start_main()+0xf5) [0x7fcc185283d5]
 23: (()+0x564555) [0x560dbdcc1555]
 NOTE: a copy of the executable, or `objdump -rdS &lt;executable&gt;` is needed to interpret this.
```

Related issues:

Related to bluestore - Bug #45765: BlueStore::_collection_list causes huge la...	**Resolved**
Related to bluestore - Bug #45994: OSD crash - in thread tp_osd_tp	**Duplicate**

History

#1 - 07/12/2019 11:59 AM - Igor Fedotov

- Project changed from Ceph to bluestore

- Category deleted (OSD)

#2 - 07/12/2019 12:42 PM - Igor Fedotov

Here is my analysis from what I've seen in your logs so far:
1) After initial issue(s) that trigger OSDs to restart (haven't investigated what was it) they are facing a **bunch** of issues with allocating additional space for bluefs. The latter attempts to get more space to recover after the initial failure - most probably to replay WAL and flush DB data to DB/main volumes. The key thing is that this allocation differs from the regular ones and tend to quite large. Which probably reals most of the subsequent issues. Ones that I could see in your logs (including ones got from the previous ceph-users posts).

1) http://tracker.ceph.com/issues/40080 - bitmap allocator returns duplicate entries (fixed in master branch and approved and awaiting for qa for nautilus). Not 100% sure since the provided logs lacks some key information but highly likely this happened for the first log you shared.
2) http://tracker.ceph.com/issues/40703 - stupid allocator might return extents of 0 bytes length. Fix is approved and pending QA. This was observed in log shared here : https://pastebin.com/yuJKcPvX. Reproduced in the lab as well.
3)From the log attached to this ticket it looks like lack of space (partially caused by high fragmentation) is observed for this specific OSD:
-31> 2019-07-11 18:47:54.676 7f9fd6424d80 1 bluefs _allocate failed to allocate 0xcd00000 on bdev 1, free 0x2300000; fallback to bdev 2
^^^ Failing to allocate ~200 MB from DB volume which has just ~35MB free.

```
-30> 2019-07-11 18:47:54.676 7f9fd6424d80  1 bluefs _allocate unable to allocate 0xcd00000 on bdev 2, free 0xffffffffffffffff; fallback to slow device expander
^^^ Failing to allocate ~214 MB from already allocated bluefs part of the main device, Failing to do that, reported free space value 0xffffffffffffffff looks suspicious but nevertheless going to take more space from main device for bluefs.



-29> 2019-07-11 18:47:54.676 7f9fd6424d80 10 bluestore(/var/lib/ceph/osd/ceph-123) _get_bluefs_size_delta blue
```

fs 35 MiB free (8.11483e-05) bluestore 741 MiB free (0.00162123), bluefs_ratio 0.0450768
   -28> 2019-07-11 18:47:54.676 7f9fd6424d80 10 bluestore(/var/lib/ceph/osd/ceph-123) allocate_bluefs_freespac
e gifting 214958080 (205 MiB)
^^ performing the allocation from the main device....
   -27> 2019-07-11 18:47:54.676 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0xcd00000/100000,0,0

   -26> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x59e00000~100000/100000,0,0

   -25> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x592b00000~200000/100000,0,0

   -24> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x5531000000~100000/100000,0,0

   -23> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x555fa00000~100000/100000,0,0

   -22> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x5ec2300000~300000/100000,0,0

```
   -20> 2019-07-11 18:47:54.677 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 allocate 0x6c11700000~300000/100000,0,0
   -19> 2019-07-11 18:47:54.677 7f9fd6424d80 -1 bluestore(/var/lib/ceph/osd/ceph-123) allocate_bluefs_freespace failed to allocate on 0xcd00000 min_size 0xcd00000 >
allocated total 0xe00000 bluefs_alloc_size 0x100000 allocated 0xe00000 available 0x 2d774000
^
```

^ failed to do that, only 14Mb is available in contiguous 1MB chunks. Totally ~760MB free is available at main device. Which is terribly low in fact!!!

```
-18> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 0(< 32 KiB) : 4095 extents
   -17> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 1(< 64 KiB) : 1559 extents
   -16> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 2(< 128 KiB) : 1025 extent
s
   -15> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 3(< 256 KiB) : 386 extents
   -14> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 4(< 512 KiB) : 341 extents
   -13> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 5(< 1 MiB) : 364 extents
   -12> 2019-07-11 18:47:54.681 7f9fd6424d80  0 fbmap_alloc 0x561e2045dd00 dump bin 6(< 2 MiB) : 68 extents
   -11> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x59e00000~100000
   -10> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x592b00000~200000
    -9> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x5531000000~100000
    -8> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x555fa00000~100000
    -7> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x5ec2300000~300000
    -6> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x5eda900000~300000
    -5> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release 0x6c11700000~300000
    -4> 2019-07-11 18:47:54.681 7f9fd6424d80 10 fbmap_alloc 0x561e2045dd00 release done
    -3> 2019-07-11 18:47:54.681 7f9fd6424d80 -1 bluefs _allocate failed to expand slow device to fit +0xcd0000
0
    -2> 2019-07-11 18:47:54.681 7f9fd6424d80 -1 bluefs _flush_range allocated: 0x0 offset: 0x0 length: 0xccf60
b1
 -1> 2019-07-11 18:47:54.688 7f9fd6424d80 -1 /home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILA
BLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1
/src/os/bluestore/BlueFS.cc: In function 'int BlueFS::_flush_range(BlueFS::FileWriter*, uint64_t, uint64_t)' t
hread 7f9fd6424d80 time 2019-07-11 18:47:54.682321
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/c
entos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1/src/os/bluestore/BlueFS.cc: 2044: abort()
^^^ aborting due to the lack of free space
```

So my general considerations/notes:
1) do OSD troubleshooting and repairing one by one - they have different issues. Do the fix for a single specific node and wait until it has started successfully. Proceeding cluster malfunction after applying some fix to all doesn't mean the fix doesn't help - you might simply get another issue.

2) revise free space at your OSDs - the existing report (just for a single OSD so far) shows very little space at both DB and main device.
3) I suggest to be careful with any massive data migrations (or preferably avoid them at all) unless you know exactly what's happening and why this will help..

#3 - 07/12/2019 04:05 PM - Brett Chancellor

Thanks for looking into Igor. That was one of the many failed SSD volumes, chosen at random. Here is some info from the first one to die (Didn't have debugging on when it died)

2019-07-11 08:13:00.966 7fc7a9b06700 0 osd.196 pg_epoch: 1352621 pg[60.32(v 1352617'911187 (1333356'908185,1352617'911187] lb 60:4db19752:::.meta%3abucket.instance%3adb_visibility_and_automation%2f2019-7-5-prod-log_file_type_oracle_sfcon_sql%3a8346a3be-abaa-4aa6-8d8a-f54818d16aef.146035057.1%3a_2oTtyanhCocW2sZKQjEgEme%3a27678:head (bitwise) local-lis/les=1352490/1352491 n=235514 ec=2213/2213 lis/c 1352490/1351508 les/c/f 1352491/1351509/0 1352489/1352490/1352490) [136,163,196]/[59,129,138] r=-1 lpr=1352490 pi=[1351508,1352490)/1 luod=0'0 lua=1352394'911185 crt=1352617'911187 active mbc={}] do_backfill primary 89225304 local 89225308
2019-07-11 08:13:00.971 7fc7bc32b700 1 bluefs _allocate failed to allocate 0x4300000 on bdev 1, free 0xc00000; fallback to bdev 2
2019-07-11 08:13:00.971 7fc7bc32b700 1 bluefs _allocate unable to allocate 0x4300000 on bdev 2, free 0xffffffffffffffff; fallback to slow device expander
2019-07-11 08:13:00.971 7fc7bc32b700 -1 bluestore(/var/lib/ceph/osd/ceph-196) allocate_bluefs_freespace failed to allocate on 0x4300000 min_size 0x4300000 > allocated total 0x1900000 bluefs_alloc_size 0x100000 allocated 0x1900000 available 0x 28c8000
2019-07-11 08:13:00.971 7fc7aab08700 0 osd.196 pg_epoch: 1352621 pg[60.8(v 1352621'819543 (1333060'816540,1352621'819543] lb 60:12631b8d:::.meta%3abucket.instance%3adb_visibility_and_automation%2f2019-5-12-prod-log_file_type_oracle_hourly_awr_text%3a8346a3be-abaa-4aa6-8d8a-f54818d16aef.126164230.1411%3a_gVWxpZjgXL1TJse8MuJNehU%3a7618:head (bitwise) local-lis/les=1352490/1352491 n=332210 ec=2213/2213 lis/c 1352490/1351448 les/c/f 1352491/1352064/0 1352523/1352523/1352490) [168,196,257]/[85,108,161] r=-1 lpr=1352523 pi=[1351448,1352523)/1 luod=0'0 lua=1352502'819541 crt=1352621'819543 active+remapped mbc={}] do_backfill primary 109664820 local 109665205
2019-07-11 08:13:00.979 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 0(< 32 KiB) : 745 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 1(< 64 KiB) : 188 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 2(< 128 KiB) : 58 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 3(< 256 KiB) : 17 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 4(< 512 KiB) : 21 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 5(< 1 MiB) : 6 extents
2019-07-11 08:13:00.993 7fc7bc32b700 0 fbmap_alloc 0x5555fd7fd300 dump bin 6(< 2 MiB) : 2 extents
2019-07-11 08:13:00.993 7fc7bc32b700 -1 bluefs _allocate failed to expand slow device to fit +0x4300000
2019-07-11 08:13:00.993 7fc7bc32b700 -1 bluefs _flush_range allocated: 0x0 offset: 0x0 length: 0x427347b
2019-07-11 08:13:01.121 7fc7a7301700 0 osd.196 pg_epoch: 1352621 pg[60.3f(v 1352609'842419 (1333048'839418,1352609'842419] lb 60:fe2b5ac9:::.meta%3abucket.instance%3adb_visibility_and_automation%2f2019-5-5-prod-log_file_type_oracle_sfcon_total%3a8346a3be-abaa-4aa6-8d8a-f54818d16aef.126164230.507%3a_EQ3kaoVN0q0czsMY73gVrpI%3a19950:head (bitwise) local-lis/les=1352490/1352491 n=301177 ec=2213/2213 lis/c 1352490/1351462 les/c/f 1352491/1351463/0 1352489/1352490/1352490) [123,196,151]/[8,26,251] r=-1 lpr=1352490 pi=[1351462,1352490)/1 luod=0'0 lua=1352392'842418 crt=1352609'842419 active mbc={}] do_backfill primary 114101245 local 114101239
2019-07-11 08:13:01.151 7fc7bc32b700 -1 /home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1/src/os/bluestore/BlueFS.cc: In function 'int BlueFS::_flush_range(BlueFS::FileWriter*, uint64_t, uint64_t)' thread 7fc7bc32b700 time 2019-07-11 08:13:01.001811
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/14.2.1/rpm/el7/BUILD/ceph-14.2.1/src/os/bluestore/BlueFS.cc: 2044: abort()

```
ceph version 14.2.1 (d555a9489eb35f84f2e1ef49b77e19da9d113972) nautilus (stable)
 1: (ceph::__ceph_abort(char const*, int, char const*, std::string const&)+0xd8) [0x5555f1d01cd0]
 2: (BlueFS::_flush_range(BlueFS::FileWriter*, unsigned long, unsigned long)+0x1daf) [0x5555f23b99af]
 3: (BlueFS::_flush(BlueFS::FileWriter*, bool)+0x10b) [0x5555f23b9b4b]
 4: (BlueRocksWritableFile::Flush()+0x3d) [0x5555f23d784d]
 5: (rocksdb::WritableFileWriter::Flush()+0x19e) [0x5555f2a05d0e]
 6: (rocksdb::WritableFileWriter::Sync(bool)+0x2e) [0x5555f2a05fee]
 7: (rocksdb::CompactionJob::FinishCompactionOutputFile(rocksdb::Status const&, rocksdb::CompactionJob::SubcompactionState*, rocksdb::RangeDelAggregator*, CompactionIterationStats*, rocksdb::Slice const*)+0xbaa) [0x5555f2a5373a]
 8: (rocksdb::CompactionJob::ProcessKeyValueCompaction(rocksdb::CompactionJob::SubcompactionState*)+0x7d0) [0x5555f2a57150]
 9: (rocksdb::CompactionJob::Run()+0x298) [0x5555f2a58618]
 10: (rocksdb::DBImpl::BackgroundCompaction(bool*, rocksdb::JobContext*, rocksdb::LogBuffer*, rocksdb::DBImpl::PrepickedCompaction*)+0xcb7) [0x5555f2897b67]
 11: (rocksdb::DBImpl::BackgroundCallCompaction(rocksdb::DBImpl::PrepickedCompaction*, rocksdb::Env::Priority)+0xd0) [0x5555f28993c0]
 12: (rocksdb::DBImpl::BGWorkCompaction(void*)+0x3a) [0x5555f289990a]
 13: (rocksdb::ThreadPoolImpl::Impl::BGThread(unsigned long)+0x264) [0x5555f2aa59c4]
 14: (rocksdb::ThreadPoolImpl::Impl::BGThreadWrapper(void*)+0x4f) [0x5555f2aa5b4f]
 15: (()+0x129dfff) [0x5555f2b32fff]
 16: (()+0x7dd5) [0x7fc7c941add5]
 17: (clone()+0x6d) [0x7fc7c82e0ead]
```

2019-07-11 08:13:01.253 7fc7bc32b700 -1 ** **Caught signal (Aborted) ***
in thread 7fc7bc32b700 thread_name:rocksdb:low0

```
ceph version 14.2.1 (d555a9489eb35f84f2e1ef49b77e19da9d113972) nautilus (stable)
 1: (()+0xf5d0) [0x7fc7c94225d0]
 2: (gsignal()+0x37) [0x7fc7c8219207]
 3: (abort()+0x148) [0x7fc7c821a8f8]
```

```
 4: (ceph::__ceph_abort(char const*, int, char const*, std::string const&)+0x19c) [0x5555f1d01d94]
 5: (BlueFS::_flush_range(BlueFS::FileWriter*, unsigned long, unsigned long)+0x1daf) [0x5555f23b99af]
 6: (BlueFS::_flush(BlueFS::FileWriter*, bool)+0x10b) [0x5555f23b9b4b]
 7: (BlueRocksWritableFile::Flush()+0x3d) [0x5555f23d784d]
 8: (rocksdb::WritableFileWriter::Flush()+0x19e) [0x5555f2a05d0e]
 9: (rocksdb::WritableFileWriter::Sync(bool)+0x2e) [0x5555f2a05fee]
 10: (rocksdb::CompactionJob::FinishCompactionOutputFile(rocksdb::Status const&, rocksdb::CompactionJob::Subco
mpactionState*, rocksdb::RangeDelAggregator*, CompactionIterationStats*, rocksdb::Slice const*)+0xbaa) [0x5555
f2a5373a]
 11: (rocksdb::CompactionJob::ProcessKeyValueCompaction(rocksdb::CompactionJob::SubcompactionState*)+0x7d0) [0
x5555f2a57150]
 12: (rocksdb::CompactionJob::Run()+0x298) [0x5555f2a58618]
 13: (rocksdb::DBImpl::BackgroundCompaction(bool*, rocksdb::JobContext*, rocksdb::LogBuffer*, rocksdb::DBImpl:
:PrepickedCompaction*)+0xcb7) [0x5555f2897b67]
 14: (rocksdb::DBImpl::BackgroundCallCompaction(rocksdb::DBImpl::PrepickedCompaction*, rocksdb::Env::Priority)
+0xd0) [0x5555f28993c0]
 15: (rocksdb::DBImpl::BGWorkCompaction(void*)+0x3a) [0x5555f289990a]
 16: (rocksdb::ThreadPoolImpl::Impl::BGThread(unsigned long)+0x264) [0x5555f2aa59c4]
 17: (rocksdb::ThreadPoolImpl::Impl::BGThreadWrapper(void*)+0x4f) [0x5555f2aa5b4f]
 18: (()+0x129dfff) [0x5555f2b32fff]
 19: (()+0x7dd5) [0x7fc7c941add5]
 20: (clone()+0x6d) [0x7fc7c82e0ead]
 NOTE: a copy of the executable, or `objdump -rdS &lt;executable&gt;` is needed to interpret this.
```

#4 - 07/12/2019 04:28 PM - Igor Fedotov

This one doesn't have enough space as well, 0xc00000 bytes as ssd, 0x28c8000 bytes at main device. See:

2019-07-11 08:13:00.971 7fc7bc32b700 1 bluefs _allocate failed to allocate 0x4300000 on bdev 1, free 0xc00000; fallback to bdev 2
2019-07-11 08:13:00.971 7fc7bc32b700 1 bluefs _allocate unable to allocate 0x4300000 on bdev 2, free 0xffffffffffffffff; fallback to slow device expander
2019-07-11 08:13:00.971 7fc7bc32b700 -1 bluestore(/var/lib/ceph/osd/ceph-196) allocate_bluefs_freespace failed to allocate on 0x4300000 min_size 0x4300000 > allocated total 0x1900000 bluefs_alloc_size 0x100000 allocated 0x1900000 available 0x 28c8000

#5 - 07/12/2019 04:30 PM - Igor Fedotov

What's behind you DB volumes - LVM or plain partition/device?

#6 - 07/12/2019 06:25 PM - Brett Chancellor

- File ceph-osd.34.log.truncated.gz added

- File ceph-osd.110.log.truncated.gz added

- File ceph-osd.44.log.truncated.gz added

LVM..

The bigger issue right now isn't the failing SSDs, it's the constantly HDD's that are constantly rebooting themselves.
Here is the HDD tree from 'ceph osd tree down'

```
-1       986.37585 root default
-460      20.00000     rack a37-45
-21       20.00000         host ceph0rdi-osd3-4-prd
51   hdd   5.00000             osd.51                  down  0.79999 1.00000
53   hdd   5.00000             osd.53                  down  0.79999 1.00000
-493      20.00000     rack f36-45
-125      20.00000         host ceph0rdi-osd3-19-prd
220   hdd   5.00000             osd.220                 down  0.79999 1.00000
-2       344.69452     rack group1
-10       16.36679         host ceph0rdi-osd1-11-prd
13   hdd   5.45560             osd.13                  down  1.00000 1.00000
-30       16.36679         host ceph0rdi-osd1-12-prd
1   hdd   5.45560              osd.1                   down  1.00000 1.00000
-31       16.36679         host ceph0rdi-osd1-13-prd
67   hdd   5.45560             osd.67                  down  1.00000 1.00000
-94       16.36679         host ceph0rdi-osd1-15-prd
47   hdd   5.45560             osd.47                  down  1.00000 1.00000
-11       16.36679         host ceph0rdi-osd1-16-prd
113   hdd   5.45560            osd.113                 down  1.00000 1.00000
-95       16.36679         host ceph0rdi-osd1-17-prd
80   hdd   5.45560             osd.80                  down  1.00000 1.00000
-96       16.50000         host ceph0rdi-osd1-18-prd
157   hdd   5.50000            osd.157                 down  1.00000 1.00000
-119      16.66553         host ceph0rdi-osd1-19-prd
61   hdd   5.55518             osd.61                  down  1.00000 1.00000
-93       16.36679         host ceph0rdi-osd1-2-prd
89   hdd   5.45560             osd.89                  down  1.00000 1.00000
-403      16.49849         host ceph0rdi-osd1-21-prd
242   hdd   5.49950            osd.242                 down  1.00000 1.00000
-5        16.36679         host ceph0rdi-osd1-3-prd
0   hdd   5.45560              osd.0                   down  1.00000 1.00000
-6        16.36679         host ceph0rdi-osd1-7-prd
83   hdd   5.45560             osd.83                  down  1.00000 1.00000
-7        16.36679         host ceph0rdi-osd1-8-prd
6   hdd   5.45560              osd.6                   down  1.00000 1.00000
-3       289.45032     rack group2
-12       16.36679         host ceph0rdi-osd2-1-prd
19   hdd   5.45560             osd.19                  down  1.00000 1.00000
26   hdd   5.45560             osd.26                  down  1.00000 1.00000
-97       10.91039         host ceph0rdi-osd2-12-prd
185   hdd   3.63680            osd.185                 down  1.00000 1.00000
-99       10.09140         host ceph0rdi-osd2-14-prd
54   hdd   3.36380             osd.54                  down  1.00000 1.00000
-100      10.91039         host ceph0rdi-osd2-16-prd
204   hdd   3.63680            osd.204                 down  1.00000 1.00000
-17       16.36679         host ceph0rdi-osd2-17-prd
32   hdd   5.45560             osd.32                  down  1.00000 1.00000
-18       16.36679         host ceph0rdi-osd2-18-prd
58   hdd   5.45560             osd.58                  down  1.00000 1.00000
112   hdd   5.45560            osd.112                 down  1.00000 1.00000
-127      16.66553         host ceph0rdi-osd2-19-prd
178   hdd   5.55518            osd.178                 down  0.98000 1.00000
-33       10.09140         host ceph0rdi-osd2-2-prd
84   hdd   3.36380             osd.84                  down  1.00000 1.00000
-400      16.50000         host ceph0rdi-osd2-28-prd
239   hdd   5.50000            osd.239                 down  0.92999 1.00000
241   hdd   5.50000            osd.241                 down  0.96999 1.00000
-34       10.63739         host ceph0rdi-osd2-3-prd
192   hdd   3.63680            osd.192                 down  1.00000 1.00000
-35       10.09140         host ceph0rdi-osd2-4-prd
138   hdd   3.36380            osd.138                 down  1.00000 1.00000
148   hdd   3.36380            osd.148                 down  1.00000 1.00000
-36       10.09140         host ceph0rdi-osd2-5-prd
55   hdd   3.36380             osd.55                  down  1.00000 1.00000
-37       10.09140         host ceph0rdi-osd2-7-prd
98   hdd   3.36380             osd.98                  down  1.00000 1.00000
-14       16.36679         host ceph0rdi-osd2-8-prd
21   hdd   5.45560             osd.21                  down  1.00000 1.00000
24   hdd   5.45560             osd.24                  down  1.00000 1.00000
-15       16.36679         host ceph0rdi-osd2-9-prd
144   hdd   5.45560            osd.144                 down  1.00000 1.00000
```

```
-4       312.23102     rack group3
-45        16.36679         host ceph0rdi-osd3-11-prd
129   hdd   5.45560             osd.129                 down  1.00000 1.00000
-101       16.36679         host ceph0rdi-osd3-14-prd
139   hdd   5.45560             osd.139                 down  1.00000 1.00000
-102       16.36679         host ceph0rdi-osd3-15-prd
44    hdd   5.45560             osd.44                  down  1.00000 1.00000
-103       16.50000         host ceph0rdi-osd3-16-prd
110   hdd   5.50000             osd.110                 down  1.00000 1.00000
119   hdd   5.50000             osd.119                 down  1.00000 1.00000
-104       16.50000         host ceph0rdi-osd3-17-prd
252   hdd   5.50000             osd.252                 down  0.96999 1.00000
-20        16.36679         host ceph0rdi-osd3-2-prd
50    hdd   5.45560             osd.50                  down  1.00000 1.00000
52    hdd   5.45560             osd.52                  down  1.00000 1.00000
-42        16.36679         host ceph0rdi-osd3-6-prd
34    hdd   5.45560             osd.34                  down  1.00000 1.00000
-22        16.36679         host ceph0rdi-osd3-7-prd
93    hdd   5.45560             osd.93                  down  1.00000 1.00000
-43        16.66553         host ceph0rdi-osd3-8-prd
124   hdd   5.55518             osd.124                 down  0.92999 1.00000
```

1. Included are logs from some of the HDDs, both are last 50k lines with heartbeatmap and debug to 20/20
2. OSD.34 - this one is times out with heartbeat issues.
3. OSD.110 - this one takes 10-15 minutes to boot then commits suicide
4. osd.44 - last 40k lines of log file, received Interrupt from kernel, not sure why

#7 - 07/12/2019 07:59 PM - Igor Fedotov

Let's keep osd.44 aside for now. For 35 & 110 please answer/do the following.
1) Check corresponding disk activity for these specific osds. Are there any (more or less massive) disk read/writes observed for them after startup?
2) set debug bluefs to 20 (and debug osd back to default) - is osd log growing fast after that. In other words is there any massive BlueFS I/O present all the time?
3) If so - please do manual compaction for one of the osd using ceph-kvstore-tool (this might take a while). Then check for the presence of suicide timeout in the log again and/or check 1) & 2) above.

Do not set debug bluefs back to 5/5 when done.

#8 - 07/12/2019 10:35 PM - Brett Chancellor

- File osd.34.bluefs.log.gz added

- File osd.110.bluefs.log.gz added

1. Info below
2. Attached last 50k lines of logs with debug_bluefs set to 20/20
3. Can you share the syntax for ceph-kvstore-tool? Is it something like ceph-kvstore-tool bluestore-kv /var/lib/ceph/osd/ceph-34 compact ?

########

1. OSD 34 ######## ====== osd.34 ======
   [block] /dev/ceph-76ef67f4-9c06-4944-9da6-aa3e150b4690/osd-block-31fcef5f-be04-4674-b781-7434460b9113
   block device /dev/ceph-76ef67f4-9c06-4944-9da6-aa3e150b4690/osd-block-31fcef5f-be04-4674-b781-7434460b9113
   db device /dev/sdf3
   devices /dev/sdd

1. Restarted osd @ 21:40 server time

1. isotat during osd boot process (60 second intervals)
   $ iostat -yxm 60 |egrep 'Device|^sdd|^sdf'
   $ iostat -yxm 60 |egrep 'Device|^sdd|^sdf'
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.02 0.00 1013.70 0.00 101.56 0.00 205.18 3.62 3.61 3.61 0.00 0.79 79.96
   sdf 0.00 25.03 96.50 7.28 1.13 0.13 24.72 0.01 0.07 0.07 0.06 0.06 0.61
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1213.68 0.00 136.25 0.00 229.90 0.83 0.68 0.68 0.00 0.68 82.88
   sdf 0.00 7.82 0.00 8.20 0.00 0.06 15.63 0.00 0.05 0.00 0.05 0.05 0.04
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1135.53 0.00 128.49 0.00 231.73 0.84 0.74 0.74 0.00 0.74 84.20
   sdf 0.00 9.23 7.53 8.38 0.03 0.07 12.70 0.00 0.04 0.04 0.04 0.04 0.06
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1098.18 0.00 124.97 0.00 233.06 0.84 0.77 0.77 0.00 0.77 84.12
   sdf 0.00 7.70 29.67 7.90 1.93 0.06 108.81 0.01 0.20 0.24 0.05 0.20 0.74
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1258.47 0.00 137.41 0.00 223.62 0.84 0.66 0.66 0.00 0.66 83.51
   sdf 0.00 7.05 15.82 7.02 0.49 0.05 48.79 0.00 0.12 0.15 0.06 0.12 0.27
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1198.12 0.00 133.33 0.00 227.90 0.84 0.70 0.70 0.00 0.70 84.36
   sdf 0.00 7.03 0.35 7.58 0.04 0.06 24.84 0.00 0.08 0.38 0.07 0.08 0.06
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.00 1254.62 0.00 139.06 0.00 227.00 0.82 0.66 0.66 0.00 0.66 82.45
   sdf 0.00 7.70 11.23 8.10 0.51 0.06 60.86 0.00 0.13 0.18 0.07 0.13 0.26
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 1.93 1.95 963.63 3.45 40.69 0.36 86.93 0.34 0.35 0.35 0.66 0.34 33.32
   sdf 0.00 10.70 179.38 24.80 0.92 0.14 10.61 0.01 0.04 0.04 0.05 0.04 0.88
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 1.95 2.73 12.15 7.80 1.44 0.72 221.35 0.02 1.15 1.52 0.57 0.90 1.81
   sdf 0.00 18.57 1.40 41.37 0.08 0.23 15.11 0.00 0.05 0.26 0.04 0.05 0.21
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdd 0.00 0.42 8.60 1.20 0.95 0.22 243.58 0.01 1.31 1.44 0.36 1.28 1.26
   sdf 0.00 9.48 0.23 12.25 0.02 0.08 16.79 0.00 0.07 0.29 0.06 0.06 0.08

#########

1. OSD 110 ######### ===== osd.110 ======
   [block] /dev/ceph-b7571325-6bca-4030-8300-146c9033a015/osd-block-d8dbe7e5-63c3-4d92-a430-baab1f73d59e
   block device /dev/ceph-b7571325-6bca-4030-8300-146c9033a015/osd-block-d8dbe7e5-63c3-4d92-a430-baab1f73d59e
   db device /dev/sdf1
   devices /dev/sdb

1. iostat while the old process is up but not responsive (5 second intervals)
   $ iostat -yxm 5 |egrep 'Device|^sdb|^sdf'
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdb 0.00 0.60 0.00 0.80 0.00 0.11 270.00 0.00 0.00 0.00 0.00 0.00 0.00
   sdf 0.00 8.60 0.00 11.20 0.00 0.08 14.14 0.00 0.04 0.00 0.04 0.04 0.04
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdb 0.00 1.00 51.40 1.20 6.15 0.18 246.36 0.04 0.68 0.69 0.17 0.68 3.58
   sdf 0.00 17.80 3.40 31.40 0.38 0.19 33.38 0.00 0.09 0.41 0.05 0.09 0.30
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdb 0.00 0.20 0.00 0.20 0.00 0.04 360.00 0.00 0.00 0.00 0.00 0.00 0.00
   sdf 0.00 2.40 0.00 2.80 0.00 0.02 14.86 0.00 0.00 0.00 0.00 0.00 0.00

1. restart process @ 21:24 server time

1. iostat during osd boot process (60 second intervals)
   $ iostat -yxm 60 |egrep 'Device|^sdb|^sdf'
   Device: rrqm/s wrqm/s r/s w/s rMB/s wMB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
   sdb 0.02 0.00 464.62 0.00 35.56 0.00 156.73 8.09 17.46 17.46 0.00 1.59 74.08
   sdf 0.00 104.25 272.07 7.72 15.71 0.44 118.19 0.07 0.25 0.26 0.11 0.25 6.88

```
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sdb               0.00     0.00 1202.00    0.00   134.31     0.00   228.84     0.84    0.70    0.70    0.00   0.70  83.78
sdf               0.00     5.08    1.28    8.75     0.16     0.05    43.18     0.00    0.08    0.32    0.05   0.08   0.08
Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
sdb               0.00     0.00 1187.82    0.00   132.38     0.00   228.25     0.84    0.71    0.71    0.00   0.71  84.10
sdf               0.00     6.85    0.02   11.62     0.00     0.07    12.72     0.00    0.04    0.00    0.04   0.04   0.04
```

#9 - 07/13/2019 03:45 AM - Igor Fedotov

Brett Chancellor wrote:

1. Info below
2. Attached last 50k lines of logs with debug_bluefs set to 20/20
3. Can you share the syntax for ceph-kvstore-tool? Is it something like ceph-kvstore-tool bluestore-kv /var/lib/ceph/osd/ceph-34 compact ?

yes, the syntax is like that

ceph-kvstore-tool bluestore-kv <path-to-osd> compact

Just to keep everybody aware - for these two osds the issue reminds me http://tracker.ceph.com/issues/36482

Wondering if you can have some custom build from sources to double check that?

#10 - 07/17/2019 01:21 PM - Igor Fedotov

- Status changed from New to 12

- Assignee set to Igor Fedotov

#11 - 07/25/2019 12:07 PM - Igor Fedotov

Here is a summary of what we've discovered during this issue troubleshooting.
1) OSDs were dying due to suicide timeout, Mass heart beat timeouts were observed before
2) manual rocks db compaction helped for a while (several hours) and the issue reappeared again
3) Most of times one could see BlueStore collection_list function (called from PG::PG::do_delete_work) in backtraces printed after suicide.
4) Pool removal task running in background was found - it had been initiated several (3?) days before the troubleshooting session. Corresponding pool had ~32M objects. initially. Presumably this was <zone>.rgw.meta pool mentioned above.
5) Doing PG listing for the pool being removed using ceph-objectstore-tool was taking 20+ minutes and returned around 400K objects. Doing the same for other large pool took ~20 seconds (compare to minutes!) while returned just twice as less entries.
6) BlueFS allocated size on inspected OSDs was around 500GB which just 150GB of it residing at SSD drives. I.e. huge spillover took place.
7) Finally the hypothesis about very slow collection listing which blocks OSDs for a while has appeared. Such listings (max 30 entries per shot) are massively issued during pool removal. Due to lack of proper tooling we were unable to measure the duration for this short listings but most probably they were still very high which triggered heartbeat timeouts and finally caused suicides.
8) The suggested workaround is to stop background removal by tricking with RocksDB - remove root PG (aka collection) entries related to the bad pool from RocksDB using ceph-kvstore-tool, e.g. ceph-kvstore-tool bluestore-kv dev/osd0 rm C 5.35_head
9) Which finally prevented the issue from reappearance.
10) Hence my understanding of the issue (I'm not covering what was the initial trigger here) is as follows:
Massive record removal from huge DB residing mainly on rotational disk might cause very slow entry enumeration. Presumably this applies to enumerations that are related to already removed records. In our case both removed records and new enumerations were for the same collection. To some degree this reminds the issue we observed for slow omap listing here:
https://github.com/ceph/ceph/pull/27627

http://tracker.ceph.com/issues/36482
Where holes in DB made by massive removal caused slow omap listing tail seek.

And since collection_list is a synchronous call corresponding OSD threads stuck for a quite while.

BlueFS read prefetch and/or keeping DB at fast storage might reduce the negative impact but IMO the proper solution would be making collection_list call asynchronous.

#12 - 09/12/2019 01:52 AM - Sa Pham

I have same issue with you. Did you solve this issue? Could you update infor about this one?

#13 - 09/12/2019 01:15 PM - Igor Fedotov

Please see my previous comment, on-site issue has been worked around the way shared there.

But please be cautious when saying that your issue is exactly the same - one can get OSD crashes due to suicide timeouts under absolutely different scenarios....

#14 - 12/05/2019 09:37 PM - Patrick Donnelly

- Status changed from 12 to New

#15 - 06/05/2020 09:54 PM - Igor Fedotov

- Status changed from New to Triaged

#16 - 06/05/2020 09:55 PM - Igor Fedotov

- Related to Bug #45765: BlueStore::_collection_list causes huge latency growth pg deletion added

#17 - 06/22/2020 08:32 AM - Igor Fedotov

- Related to Bug #45994: OSD crash - in thread tp_osd_tp added

Files

ceph-osd.123.log.truncated.gz	71.8 KB	07/11/2019	Brett Chancellor
ceph-osd.34.log.truncated.gz	515 KB	07/12/2019	Brett Chancellor
ceph-osd.110.log.truncated.gz	170 KB	07/12/2019	Brett Chancellor
ceph-osd.44.log.truncated.gz	318 KB	07/12/2019	Brett Chancellor
osd.34.bluefs.log.gz	505 KB	07/12/2019	Brett Chancellor
osd.110.bluefs.log.gz	486 KB	07/12/2019	Brett Chancellor