Bitmap allocator return duplicate entries which cause interval_set assert

05/30/2019 08:41 AM - Xiaoxi Chen

Status: Resolved
Priority: Urgent
Assignee: Igor Fedotov
Category:
Target version:
Source: Community (dev)
Tags:
Backport: mimic, luminous, nautilus
Regression: No
Severity: 3 - minor
Reviewed:

Description

0x941300000~200000 were returned several times by allocator, later when allocate_bluefs_freespace give duplicate entries to interval_set, interval_set assert.

--- STACK ---

(gdb) bt
#0 0x00005631fb8551f3 in __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1 0x00005631fc027de0 <interval_set<unsigned long, std::map<unsigned long, unsigned long, std::less<unsigned long>, std::allocator<std::pair<unsigned long const, unsigned long> > > >::insert(unsigned long, unsigned long*, unsigned long*)::__PRETTY_FUNCTION__> "void interval_set<T, Map>::insert(T, T, T*, T*) [with T = long unsigned int; Map = std::map<long unsigned int, long unsigned int>]") at ./src/common/assert.cc:73
#2 0x00005631fb12b9bb in ceph::__ceph_assert_fail (ctx=...) at ./src/common/assert.cc:78
#3 0x00005631fb459301 in interval_set<unsigned long, std::map<unsigned long, unsigned long, std::less<unsigned long>, std::allocator<std::pair<unsigned long const, unsigned long> > > >::insert (this=0x5632071af360, start=<optimized out>, len=2097152, pstart=<optimized out>, plen=<optimized out>) at ./src/include/interval_set.h:490
#4 0x00005631fb6d935e in BlueStore::allocate_bluefs_freespace (this=0x5632071af000, size=<optimized out>) at ./src/os/bluestore/BlueStore.cc:5594
#5 0x00005631fb7fd03b in BlueFS::_allocate (this=this@entry=0x5632088ace58, id=<optimized out>, len=len@entry=152453959, node=0x5632088ace58) at ./src/os/bluestore/BlueFS.cc:2448
#6 0x00005631fb7ff71c in BlueFS::flush (this=0x5632088ace58) at ./src/os/bluestore/BlueFS.cc:2255
#7 0x00005631fb81ed6d in BlueRocksWritableFile::Flush (this=<optimized out>) at ./src/os/bluestore/BlueRocksEnv.cc:209
#8 0x00005631fb82ae9 in rocksdb::WritableFileWriter::Flush (this=<optimized out>) at ./src/rocksdb/util/file_reader_writer.cc:314
2019-05-30 01:16:01.303 7f8833480f00 10 fbmap_alloc 0x55a3813f5200 allocate 0x941300000~200000/100000,0,0
2019-05-30 01:16:01.295 7f8833480f00 10 fbmap_alloc 0x55a3813f5200 allocate 0x941300000~200000/100000,0,0
2019-05-30 01:16:01.295 7f8833480f00 10 fbmap_alloc 0x55a3813f5200 allocate 0x941300000~200000/100000,0,0
2019-05-30 01:16:01.291 7f8833480f00 10 fbmap_alloc 0x55a3813f5200 allocate 0x941300000~200000/100000,0,0
Related issues:
Related to bluestore - Bug #37282: rocksdb: submit_transaction_sync error: Co...
Need More Info
Copied to bluestore - Backport #40422: luminous: Bitmap allocator return dupl...
Resolved
Copied to bluestore - Backport #40423: mimic: Bitmap allocator return duplica...
Resolved
Copied to bluestore - Backport #40424: nautilus: Bitmap allocator return dupl...
Resolved

History
#1 - 05/30/2019 08:42 AM - Xiaoxi Chen
Full log with debug_bluestore=30, debug_bluefs=10 are available , but too big to upload.

#2 - 05/30/2019 09:24 AM - Igor Fedotov
Xiaoxi, this is v14.2.0, right?
14.2.1 & .2 have some bmap allocator's fixes that you might want to apply:
https://github.com/ceph/ceph/pull/27740
https://github.com/ceph/ceph/pull/27138

#3 - 05/30/2019 09:31 AM - Xiaoxi Chen
@Igor,
I noticed that during debugging, however, 14.2.1 crashed the same way.

actually Stupid allocator crashed in another way where it assert for (len > 0) and which lead me to the two changes you are pointing, especially the uint32_t->64.
I do see some very large directory in our fs…however the rebuilt-osd crashed again which bring down 8 PGs and as well as the filesystem….sucks..

It seems related with a huge allocation from bluefs
2019-05-30 01:16:01.223 7f8833480f00 10 bluefs _expand_slow_device expanding slow device by 0x679300000

and my guess is in somewhere we allocate something more than 4GB(like 8GB) and the higher 32bit was truncated in bluestore_interval_t, which make the len to 0?

Changing bluestore_bluefs_gift_ratio to 0.002 which make the allocation from 25GB to 2.5GB mitigate the issue at the moment

Managed to reproduce the same by replaying the log. WIP on the fix.
<table>
<thead>
<tr>
<th>Files</th>
<th>Size</th>
<th>Date</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>osd_87_debug_osd_10.tar.gz</td>
<td>334 KB</td>
<td>05/30/2019</td>
<td>Xiaoxi Chen</td>
</tr>
</tbody>
</table>