bluestore - Bug #38363

Failure in assert when calling: ceph-volume lvm prepare --bluestore --data /dev/sdg

02/18/2019 01:41 PM - Rainer Krienke

Status:	Need More Info	**% Done:**	0%
Priority:	Normal		
Assignee:	Adam Kupczyk		
Category:			
Target version:			
Source:	other	**Reviewed:**	
Tags:	Ubuntu 18.04.2,osd	**Affected Versions:**	v13.2.4
Backport:		**ceph-qa-suite:**	
Regression:	No	**Pull request ID:**	
Severity:	2 - major	**Crash signature:**	

Description

I run Ubuntu 18.04 and and ceph version 13.2.4-1bionic from this repo: https://download.ceph.com/debian-mimic.

When I try to create a new bluestore osd on several 4TB disks I get an error I first thought was related to http://tracker.ceph.com/issues/15386_(read_fsid unparsable uuid) . However a user cephs user list gave me a hint that in my error log I posted an assertion failure is the real problem not the _read_fsid unparsable uuid message, So I created this new bug report. The same also happens when I omit the --bluestore option.

So here is the complete log for a run of ceph-volume to create an osd which fails reproducebly. I also tried several different devices but the result was always the same:

1. ceph-volume lvm prepare --bluestore --data /dev/sdg

Running command: /usr/bin/ceph-authtool --gen-print-key
Running command: /usr/bin/ceph --cluster ceph --name client.bootstrap-osd --keyring /var/lib/ceph/bootstrap-osd/ceph.keyring -i - osd new a87c3a87-cf22-41df-af4b-c971ed4c0e1a
Running command: /sbin/vgcreate --force --yes ceph-22a3d361-78b5-40b4-8af3-74b1efe1b65a /dev/sdg
stdout: Physical volume "/dev/sdg" successfully created.
stdout: Volume group "ceph-22a3d361-78b5-40b4-8af3-74b1efe1b65a" successfully created
Running command: /sbin/lvcreate --yes -l 100%FREE -n osd-block-a87c3a87-cf22-41df-af4b-c971ed4c0e1a ceph-22a3d361-78b5-40b4-8af3-74b1efe1b65a
stdout: Logical volume "osd-block-a87c3a87-cf22-41df-af4b-c971ed4c0e1a" created.
Running command: /usr/bin/ceph-authtool --gen-print-key
Running command: /bin/mount -t tmpfs tmpfs /var/lib/ceph/osd/ceph-0
--> Absolute path not found for executable: restorecon
--> Ensure $PATH environment variable contains common executable locations
Running command: /bin/chown -h ceph:ceph /dev/ceph-22a3d361-78b5-40b4-8af3-74b1efe1b65a/osd-block-a87c3a87-cf22-41df-af4b-c971ed4c0e1a
Running command: /bin/chown -R ceph:ceph /dev/dm-8
Running command: /bin/ln -s /dev/ceph-22a3d361-78b5-40b4-8af3-74b1efe1b65a/osd-block-a87c3a87-cf22-41df-af4b-c971ed4c0e1a /var/lib/ceph/osd/ceph-0/block
Running command: /usr/bin/ceph --cluster ceph --name client.bootstrap-osd --keyring /var/lib/ceph/bootstrap-osd/ceph.keyring mon getmap -o /var/lib/ceph/osd/ceph-0/activate.monmap
stderr: got monmap epoch 1
Running command: /usr/bin/ceph-authtool /var/lib/ceph/osd/ceph-0/keyring --create-keyring --name osd.0 --add-key AQAQtGpcjkxOMxAARlPykBaxHWqIyndvjTMNuQ==
stdout: creating /var/lib/ceph/osd/ceph-0/keyring
added entity osd.0 auth auth(auid = 18446744073709551615 key=AQAQtGpcjkxOMxAARlPykBaxHWqIyndvjTMNuQ== with 0 caps)
Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0/keyring
Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0/
Running command: /usr/bin/ceph-osd --cluster ceph --osd-objectstore bluestore --mkfs -i 0 --monmap /var/lib/ceph/osd/ceph-0/activate.monmap --keyfile - --osd-data /var/lib/ceph/osd/ceph-0/ --osd-uuid a87c3a87-cf22-41df-af4b-c971ed4c0e1a --setuser ceph --setgroup ceph
stderr: 2019-02-18 14:33:07.093 7fb9508d5240 -1 bluestore(/var/lib/ceph/osd/ceph-0/) *read_fsid unparsable uuid*

stderr: /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: In function 'virtual int KernelDevice::read(uint64_t, uint64_t, ceph::bufferlist, IOContext*, bool)' thread 7fb9508d5240 time 2019-02-18 14:33:07.155877*
stderr: /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: 821: FAILED assert((uint64_t)r len)
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
*stderr: 1: (ceph::*_ceph_assert_fail(char const*, char const*, int, char const*)+0x102) [0x7fb947cf53e2]
stderr: 2: (()+0x26d5a7) [0x7fb947cf55a7]
stderr: 3: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 4: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 5: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 6: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 7: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 8: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 9: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 10: (main()+0x4222) [0x55a21df85462]
stderr: 11: (*_libc_start_main()+0xe7) [0x7fb9452b7b97]*
stderr: 12: (_start()+0x2a) [0x55a21e04e95a]
stderr: NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
stderr: 2019-02-18 14:33:07.157 7fb9508d5240 -1 /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: In function 'virtual int KernelDevice::read(uint64_t, uint64_t, ceph::bufferlist, IOContext*, bool)' thread 7fb9508d5240 time 2019-02-18 14:33:07.155877*
stderr: /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: 821: FAILED assert((uint64_t)r len)
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
*stderr: 1: (ceph::*_ceph_assert_fail(char const*, char const*, int, char const*)+0x102) [0x7fb947cf53e2]
stderr: 2: (()+0x26d5a7) [0x7fb947cf55a7]
stderr: 3: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 4: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 5: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 6: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 7: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 8: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 9: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 10: (main()+0x4222) [0x55a21df85462]
stderr: 11: (*_libc_start_main()+0xe7) [0x7fb9452b7b97]*
stderr: 12: (_start()+0x2a) [0x55a21e04e95a]
stderr: NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
stderr: -25> 2019-02-18 14:33:07.093 7fb9508d5240 -1 bluestore(/var/lib/ceph/osd/ceph-0/) _read_fsid unparsable uuid
stderr: 0> 2019-02-18 14:33:07.157 7fb9508d5240 -1 /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: In function 'virtual int KernelDevice::read(uint64_t, uint64_t, ceph::bufferlist, IOContext*, bool)' thread 7fb9508d5240 time 2019-02-18 14:33:07.155877*
stderr: /build/ceph-13.2.4/src/os/bluestore/KernelDevice.cc: 821: FAILED assert((uint64_t)r == len)
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
*stderr: 1: (ceph::*_ceph_assert_fail(char const*, char const*, int, char const*)+0x102) [0x7fb947cf53e2]
stderr: 2: (()+0x26d5a7) [0x7fb947cf55a7]
stderr: 3: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 4: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 5: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 6: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 7: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 8: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 9: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 10: (main()+0x4222) [0x55a21df85462]
stderr: 11: (*_libc_start_main()+0xe7) [0x7fb9452b7b97]*
stderr: 12: (_start()+0x2a) [0x55a21e04e95a]
stderr: NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
*stderr: *** Caught signal (Aborted) ****
stderr: in thread 7fb9508d5240 thread_name:ceph-osd
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
stderr: 1: (()+0x92aa40) [0x55a21e5e5a40]
stderr: 2: (()+0x12890) [0x7fb9463f9890]
stderr: 3: (gsignal()+0xc7) [0x7fb9452d4e97]
stderr: 4: (abort()+0x141) [0x7fb9452d6801]
stderr: 5: (ceph::_ceph_assert_fail(char const, char const*, int, char const*)+0x250) [0x7fb947cf5530]
stderr: 6: (()+0x26d5a7) [0x7fb947cf55a7]

stderr: 7: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 8: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 9: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 10: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 11: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 12: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 13: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 14: (main()+0x4222) [0x55a21df85462]
stderr: 15: (*__libc_start_main()+0xe7) [0x7fb9452b7b97]*
stderr: 16: (_start()+0x2a) [0x55a21e04e95a]
stderr: 2019-02-18 14:33:07.157 7fb9508d5240 -1 ***Caught signal (Aborted)***
stderr: in thread 7fb9508d5240 thread_name:ceph-osd
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
stderr: 1: (()+0x92aa40) [0x55a21e5e5a40]
stderr: 2: (()+0x12890) [0x7fb9463f9890]
stderr: 3: (gsignal()+0xc7) [0x7fb9452d4e97]
stderr: 4: (abort()+0x141) [0x7fb9452d6801]
stderr: 5: (ceph::_ceph_assert_fail(char const, char const*, int, char const*)+0x250) [0x7fb947cf5530]
stderr: 6: (()+0x26d5a7) [0x7fb947cf55a7]
stderr: 7: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 8: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 9: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 10: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 11: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 12: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 13: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 14: (main()+0x4222) [0x55a21df85462]
stderr: 15: (*__libc_start_main()+0xe7) [0x7fb9452b7b97]*
stderr: 16: (_start()+0x2a) [0x55a21e04e95a]
stderr: NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
stderr: 0> 2019-02-18 14:33:07.157 7fb9508d5240 -1 ***Caught signal (Aborted) ****
stderr: in thread 7fb9508d5240 thread_name:ceph-osd
stderr: ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
stderr: 1: (()+0x92aa40) [0x55a21e5e5a40]
stderr: 2: (()+0x12890) [0x7fb9463f9890]
stderr: 3: (gsignal()+0xc7) [0x7fb9452d4e97]
stderr: 4: (abort()+0x141) [0x7fb9452d6801]
*stderr: 5: (ceph::*_ceph_assert_fail(char const*, char const*, int, char const*)+0x250) [0x7fb947cf5530]
stderr: 6: (()+0x26d5a7) [0x7fb947cf55a7]
stderr: 7: (KernelDevice::read(unsigned long, unsigned long, ceph::buffer::list*, IOContext*, bool)+0x4a7) [0x55a21e5d4817]
stderr: 8: (BlueFS::_read(BlueFS::FileReader*, BlueFS::FileReaderBuffer*, unsigned long, unsigned long, ceph::buffer::list*, char*)+0x435) [0x55a21e5945c5]
stderr: 9: (BlueFS::_replay(bool, bool)+0x214) [0x55a21e59a434]
stderr: 10: (BlueFS::mount()+0x1f1) [0x55a21e59ec81]
stderr: 11: (BlueStore::_open_db(bool, bool)+0x17cd) [0x55a21e4c504d]
stderr: 12: (BlueStore::mkfs()+0x805) [0x55a21e4f5fe5]
stderr: 13: (OSD::mkfs(CephContext*, ObjectStore*, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, uuid_d, int)+0x1b0) [0x55a21e09e480]
stderr: 14: (main()+0x4222) [0x55a21df85462]
stderr: 15: (__libc_start_main()+0xe7) [0x7fb9452b7b97]
stderr: 16: (_start()+0x2a) [0x55a21e04e95a]
stderr: NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
--> Was unable to complete a new OSD, will rollback changes
Running command: /usr/bin/ceph --cluster ceph --name client.bootstrap-osd --keyring /var/lib/ceph/bootstrap-osd/ceph.keyring osd purge-new osd.0 --yes-i-really-mean-it
stderr: purged osd.0
--> RuntimeError: Command failed with exit code 250: /usr/bin/ceph-osd --cluster ceph --osd-objectstore bluestore --mkfs -i 0 --monmap /var/lib/ceph/osd/ceph-0/activate.monmap --keyfile - --osd-data /var/lib/ceph/osd/ceph-0/ --osd-uuid a87c3a87-cf22-41df-af4b-c971ed4c0e1a --setuser ceph --setgroup ceph

History

#1 - 02/28/2019 03:16 PM - Sage Weil

- Status changed from New to Need More Info

- Priority changed from Normal to High

Can you reproduce this with debug_bluestore=20, debug_bluefs=20, debug_bdev=20?

Thanks!

#2 - 03/01/2019 07:56 AM - Rainer Krienke

I tried but the output of ceph-volume remains the same....

I added this to /etc/ceph/ceph.conf on my testing node:

[osd]
debug_bluestore=20
debug_bluefs=20
debug_bdev=20

and then called

1. ceph-volume lvm prepare --bluestore --data /dev/sdg

However the output is exactly 135 in lines which is identical to the output I initially posted.
On this ceph node only a mon and mgr daemon are running, obviously no osd. I restarted the mon daemon just to be sure, but it did not help either.

Am I missing something?

#3 - 03/05/2019 11:46 AM - Rainer Krienke

- File ceph-osd.0.log added

I finally found the extended debug log in /var/log/ceph/ceph-osd.0.log. I attached the log output file (44k) to this bug report.

#4 - 03/14/2019 09:55 AM - Rainer Krienke

- File ceph-osd-err-16.04-luminous.txt added

- File ceph-osd-err-SLES12SP3-ses5.txt added

I tested more with exactly the same hardware (PowerEdge R730xd). I tried to setup ceph luminous on Ububntu 16.04 and also was unable to create OSDs. I uploaded a debug file from this session named ceph-osd-err-16.04-luminous.txt.

Next I tried SUSE SES5 server which is based upon ceph luminous and SLES12SP3. Again I see problems with the creation of OSDs. However since this setup was done via DeepSea (salt based deploy tool) it automatically tried to create OSDs on all available disks where I before had only tried one or another of the available disks. Since the DELL R730 server has an internal flash medium called "DELL IDSDM" of 14,9GB in size, deepsea also tried to create an OSD on this disk. The result was more differentiated.

This time I found that DeepSea had successfully created OSDs on the 14,9GB disk but was also unable to create any OSD on the default 4TB SAS disks of type SEAGATE ST4000NM0295. I also attached a log file for OSD creation on one of the 4TB disks: ceph-osd-err-SLES12SP3-ses5.txt

So for me it looks like this bug has something to do with the 4TB disks or driver.

Here is the hwinfo --disk output for one of the 4TB Seagate disks perhaps this helps:

185: SCSI 0f.0: 10600 Disk
[Created at block.245]
Unique ID: FpWM.nv7AYwrn0g0
Parent ID: B35A.Nih9v7J8hJ4
SysFS ID: /class/block/sdp
SysFS BusID: 0:0:15:0
SysFS Device Link: /devices/pci0000:00/0000:00:01.0/0000:02:00.0/host0/port-0:0/expander-0:0/port-0:0:15/end_device-0:0:15/target0:0:15/0:0:15:0
Hardware Class: disk
Model: "SEAGATE ST4000NM0295"
Vendor: "SEAGATE"
Device: "ST4000NM0295"
Revision: "DT31"
Serial ID: "ZC1465ZC"
Driver: "mpt3sas", "sd"
Driver Modules: "mpt3sas", "sd_mod"
Device File: /dev/sdp (/dev/sg15)
Device Files: /dev/sdp, /dev/disk/by-id/scsi-35000c50094d3c20f, /dev/disk/by-id/scsi-SSEAGATE_ST4000NM0295_ZC1465ZC, /dev/disk/by-id/wwn-0x5000c50094d3c20f, /dev/disk/by-path/pci-0000:02:00.0-sas-exp0x500056b300f237ff-phy17-lun-0
Device Number: block 8:240-8:255 (char 21:15)
Geometry (Logical): CHS 486401/255/63
Size: 7814037168 sectors a 512 bytes
Capacity: 3726 GB (4000787030016 bytes)
Config Status: cfg=no, avail=yes, need=no, active=unknown
Attached to: #25 (Serial Attached SCSI controller)

#5 - 03/26/2019 07:44 AM - Rainer Krienke

I think I found the problem or perhaps the bug in ceph, At least I found a way to work around it....

Trying to create an osd using ceph-volume I noticed that I received the following message on the dell iDRAC console for the host:

print_req_error: protection error, dev sdg, sector 3750735880

So I read about protection information and came acrosss the package sg3-utils. Running sg_readcap on the disk yields this:

$ sg_readcap --long /dev/sdg
Read Capacity results:
Protection: prot_en=1, p_type=1, p_i_exponent=0 [type 2 protection]
Logical block provisioning: lbpme=0, lbprz=0
Last logical block address=7814037167 (0x1d1c0beaf), Number of logical blocks=7814037168
Logical block length=512 bytes
Logical blocks per physical block exponent=0
Lowest aligned logical block address=0
Hence:
Device size: 4000787030016 bytes, 3815447.8 MiB, 4000.79 GB

The important information is the line starting with "Protection:" and "prot_en=1" and "p_type=1"
This information says that this disk keeps some additional bytes for protection information (PI) for each logical 512byte block (see man sg_format). It

seems that exactly this protection capability of the disk is causing trouble to ceph_volume when creating an OSD.

So ran sg_format to disable this PI on the disk:
$ sg_format -vv --format --pfu=0 /dev/sdg

For this 4TB disk it took about 10hours to complete. Afterwards sg_readcap shows:
...
Protection: prot_en=0, p_type=0, p_i_exponent=0
...

And now a run of ceph-volume lvm prepare --bluestore --data /dev/sdg runs without problems, The OSD has successfully been created.

However even with PI enabled for the original disk it is no problem to install eg linux on these type of disks only OSD creation fails.

So my question is if this behaviour is a bug in ceph that should be fixed or if it is intended behaviour for whatever reason and I really have to reformat all my disks with sg_format because ceph does intentionally **not** support PI for OSD disks?

#6 - 03/28/2019 11:29 PM - Neha Ojha

- Assignee set to Adam Kupczyk

#7 - 04/30/2019 10:07 AM - Adam Kupczyk

SCSI defines data protection in SBC-3, "4.22 Protection Information Model".
(http://www.t10.org/drafts.htm#SBC_Family , paywalled).
This gives extra bytes per sector (called Data Integrity Field) for control of data consistency.

This functionality has been implemented in linux.
https://www.landley.net/kdocs/ols/2008/ols2008v2-pages-151-156.pdf

I have looked up kernel used in Ubuntu 18.04 git://kernel.ubuntu.com/ubuntu/ubuntu-bionic.git for working of DIF in mode 2 protection.
Inspection of
./include/linux/bio.h: bip_get_seed()
./drivers/scsi/sd_dif.c: sd_dif_complete()
./block/t10-pi.c: t10_pi_generate(), t10_pi_verify()
convinces me that in protection mode 2 DIF field is used to contain logical sector position.

From that I imagine that logical sector position may have changed.
I think this can happen when two condition are met:
1) we changed partitioning, so sector numbering scheme is changed
2) we read data from sector that we have not written before

When I do "ceph-osd --mkfs" my two first reads/writes are:

#0 KernelDevice::write (this=0x555557186a80, off=4096, bl=..., buffered=false)
at /home/adam/ceph-4/src/os/bluestore/KernelDevice.cc:713
#1 0x0000555555e29cb5 in BlueFS::_write_super (this=this@entry=0x555556d9a600)
at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:526
#2 0x0000555555e3ead9 in BlueFS::mkfs (this=0x555556d9a600, osd_uuid=...) at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:374
#3 0x0000555555d5a124 in BlueStore::_open_db (this=this@entry=0x55555717e000, create=create@entry=true,
to_repair_db=to_repair_db@entry=false) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5080
#4 0x0000555555d8b0c5 in BlueStore::mkfs (this=0x55555717e000) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5686
#5 0x00005555559330d0 in OSD::mkfs (cct=0x555556d84900, store=0x55555717e000, dev=..., fsid=..., whoami=0)
at /home/adam/ceph-4/src/osd/OSD.cc:1700
#6 0x000055555581b4d2 in main (argc=<optimized out>, argv=<optimized out>) at /home/adam/ceph-4/src/ceph_osd.cc:335

#0 KernelDevice::read (this=0x555557186a80, off=4096, len=4096, pbl=0x7fffffffaf20, ioc=0x555557245dc0, buffered=false)
at /home/adam/ceph-4/src/os/bluestore/KernelDevice.cc:806
#1 0x0000555555e29798 in BlueFS::_open_super (this=this@entry=0x555556d9a600)

at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:543
#2 0x0000555555e33b58 in BlueFS::mount (this=0x555556d9a600) at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:430
#3 0x0000555555d5a12d in BlueStore::_open_db (this=this@entry=0x55555717e000, create=create@entry=true,
to_repair_db=to_repair_db@entry=false) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5082
#4 0x0000555555d8b0c5 in BlueStore::mkfs (this=0x55555717e000) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5686
#5 0x00005555559330d0 in OSD::mkfs (cct=0x555556d84900, store=0x55555717e000, dev=..., fsid=..., whoami=0)
at /home/adam/ceph-4/src/osd/OSD.cc:1700
#6 0x000055555581b4d2 in main (argc=<optimized out>, argv=<optimized out>) at /home/adam/ceph-4/src/ceph_osd.cc:335

and

#0 KernelDevice::read (this=0x555557187180, off=1048576, len=1048576, pbl=0x555557149890, ioc=0x55555725c680, buffered=true)
at /home/adam/ceph-4/src/os/bluestore/KernelDevice.cc:806
#1 0x0000555555e28e85 in BlueFS::_read (this=this@entry=0x555556d9a600, h=h@entry=0x555557149880, buf=buf@entry=0x555557149888,
off=0, len=<optimized out>, outbl=outbl@entry=0x7fffffffaea0, out=0x0) at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:1107
#2 0x0000555555e2f444 in BlueFS::_replay (this=this@entry=0x555556d9a600, noop=noop@entry=false,
to_stdout=to_stdout@entry=false) at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:596
#3 0x0000555555e33c91 in BlueFS::mount (this=0x555556d9a600) at /home/adam/ceph-4/src/os/bluestore/BlueFS.cc:440
#4 0x0000555555d5a12d in BlueStore::_open_db (this=this@entry=0x55555717e000, create=create@entry=true,
to_repair_db=to_repair_db@entry=false) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5082
#5 0x0000555555d8b0c5 in BlueStore::mkfs (this=0x55555717e000) at /home/adam/ceph-4/src/os/bluestore/BlueStore.cc:5686
#6 0x00005555559330d0 in OSD::mkfs (cct=0x555556d84900, store=0x55555717e000, dev=..., fsid=..., whoami=0)
at /home/adam/ceph-4/src/osd/OSD.cc:1700
#7 0x000055555581b4d2 in main (argc=<optimized out>, argv=<optimized out>) at /home/adam/ceph-4/src/ceph_osd.cc:335

This second read is not preceded by any write, and it looks very similarily to provided error callstack.

TEMPORARY HYPOTHESIS:
We fail because we read data before write.

I have no access to SCSI device that can enforce protection mode 2.
I cannot continue investigation.

#8 - 05/30/2019 02:08 PM - Josh Durgin

- Priority changed from High to Normal

Files

ceph-osd.0.log	43.3 KB	03/05/2019	Rainer Krienke
ceph-osd-err-16.04-luminous.txt	15.3 KB	03/14/2019	Rainer Krienke
ceph-osd-err-SLES12SP3-ses5.txt	77.5 KB	03/14/2019	Rainer Krienke