RADOS - Bug #50659

Segmentation fault under Pacific 16.2.1 when using a custom crush location hook

05/05/2021 02:21 PM - Andrew Davidoff

Status:	Resolved	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Adam Kupczyk		
Category:			
Target version:			
Source:	Community (user)	**Affected Versions:**	v16.2.1
Tags:		**ceph-qa-suite:**	
Backport:	pacific	**Component(RADOS):**	OSD
Regression:	No	**Pull request ID:**	43944
Severity:	2 - major	**Crash signature (v1):**	
Reviewed:		**Crash signature (v2):**	

Description

I feel like if this wasn't somehow just my problem, there'd be an issue open on it already, but I'm not seeing one, and I feel like I've dug about as deep as I can without checking in with you all.

I was testing an upgrade (via ceph orch with cephadm) from 15.2.9 to 16.2.1 and found that my OSDs were crashing with a segmentation fault on start up under 16.2.1. A relevant snippet of the output in the logs is:

```
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]: debug      0> 2021-05-04T15:
40:07.914+0000 7f4e61e54080 -1 *** Caught signal (Segmentation fault) **
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  in thread 7f4e61e54080 thre
ad_name:ceph-osd
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  ceph version 16.2.1 (afb906
1ab4117f798c858c741efa6390e48ccf10) pacific (stable)
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  1: /lib64/libpthread.so.0(+
0x12b20) [0x7f4e5fbbbb20]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  2: /lib64/libc.so.6(+0x9a3d
a) [0x7f4e5e8863da]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  3: (SubProcess::add_cmd_arg
(char const*)+0x4c) [0x56504e693b2c]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  4: (SubProcess::add_cmd_arg
s(char const*, ...)+0x75) [0x56504e693cc5]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  5: (ceph::crush::CrushLocat
ion::update_from_hook()+0x2d4) [0x56504e883304]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  6: (ceph::crush::CrushLocat
ion::init_on_startup()+0x3f5) [0x56504e884455]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  7: (global_init(std::map<st
d::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::__cxx11::basic
_string<char, std::char_traits<char>, std::allocator<char> >, std::less<std::__cxx11::basic_string
<char, std::char_traits<char>, std::al
locator<char> > >, std::allocator<std::pair<std::__cxx11::basic_string<char, std::char_traits<char
>, std::allocator<char> > const, std::__cxx11::basic_string<char, std::char_traits<char>, std::all
ocator<char> > > > > const*, std::vector<char const*, std::allocator<char const*> >&, unsigned int
, code_environment_t, int, bool)+0xcd
1) [0x56504e5305b1]
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  8: main()
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  9: __libc_start_main()
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  10: _start()
May 04 15:40:07 02.ceph-kubernetes.dev.lax1.REDACTED.net bash[32744]:  NOTE: a copy of the executa
ble, or `objdump -rdS <executable>` is needed to interpret this.
```

If I remove my custom crush location hook configuration (i.e. do not specify one), the OSD can start successfully, as the following logic gets triggered which shortcuts whatever is blowing up:

```
int CrushLocation::update_from_hook()
{
  if (cct->_conf->crush_location_hook.length() == 0)
    return 0;
```

Best I can tell, this segfault is happening before my script (simple bash, which I can run inside the container manually just fine) is ever executed. If I change it to something that I know should execute fine, like /bin/ls (even though this won't create reasonable output for the location), I get the same segfault in seemingly the same place, and best I can tell, the alternate executable I'm testing with (in this case /bin/ls) is never run, same as when my script is specified.

I am not a c++ developer but based on my understanding of what I think is relevant code, and web searches, I think the segfault might be coming from a push_back happening on the cmd_args vector in add_cmd_arg. I could be totally wrong about that, but that's where I'm at. strace indicated the SIGSEGV was of type SEGV_MAPERR and I believe the address in question was 0x3 (I no longer have this output handy, however).

I am running all ceph daemons in containers as pulled from docker hub. They are running under docker on Ubuntu 20.04 systems. I have tried docker 19.03.8-0ubuntu1.20.04 and 19.03.8-0ubuntu1.20.04.2, and kernels 5.4.0-42-generic, 5.4.0-71-generic, and HWE 5.8.0-49-generic. The dev cluster I was testing the upgrade in is built from KVM instances, but I was able to reproduce this with a baremetal as well.

I am attaching the full logs of such a failed start.

Please let me know what else I can provide to help here. Thanks.

Related issues:

Copied to RADOS - Backport #53480: pacific: Segmentation fault under Pacific ...	**Resolved**

History

#1 - 05/05/2021 02:29 PM - Andrew Davidoff

I forgot to add that I tried to diff code I thought was relevant between tags v15.2.9 and v16.2.1 and thought I saw some win32 related changes that looked "close" to the potentially problematic code, I don't think I saw anything that stood out as code changes that would have broken this, which makes me wonder if it was a compiler issue - which I only suggest because I did find bug reports for segfaults on push_back that seemed to be caused by some buggy compilers, but I know that may be a long shot. I don't normally suggest it's the compiler's fault.

#2 - 05/07/2021 09:57 PM - Neha Ojha

- Status changed from New to Need More Info

Is it possible for you to capture a coredump? Did the same crush_location_hook work fine on your 15.2.9 cluster?

#3 - 05/07/2021 11:39 PM - Andrew Davidoff

- File core.ceph-osd.1620430233.gz added

I have attached a coredump. This hook works fine in 15.2.9. I can also run it fine manually from inside a launched OSD container under 16.2.1. I don't think the OSD is actually getting to the point of execing the location hook. Please let me know if I can provide anything else.

#4 - 05/25/2021 08:20 PM - Andrew Davidoff

Here's a bit more info that may be useful. Only because it's a volume already exported to the container out of the box, the crush location hook I am using lives under what the container sees as /var/log/ceph (on the host it's /var/log/ceph/$FSID). Maybe something about that location is problematic? Though as I noted earlier, trying something under /bin, which is part of the container, produced the same results.

#5 - 06/21/2021 08:50 PM - Andrew Davidoff

FYI I tried with ceph/daemon-base:master-24e1f91-pacific-centos-8-x86_64 (the latest non-devel build at this time) just to see if somehow something was different there, since that build was newer (even though it should be and is still 16.2.4), and the problem persists there too.

#6 - 07/01/2021 09:35 PM - Neha Ojha

- Priority changed from Normal to Urgent

#7 - 07/08/2021 02:43 PM - Andrew Davidoff

I just wanted to note that I see the status is listed as "Need More Info", but I think I have provided everything I have been asked for, and anything I can think of additionally. This is not me being a nag, just wanted to be clear about my perspective on the status of this ticket as it pertains to my input.

#8 - 07/09/2021 10:19 PM - Neha Ojha

- Status changed from Need More Info to New

#9 - 07/19/2021 02:20 PM - Andrew Davidoff

I saw that 16.2.5 was released. Though I didn't expect it to address this issue, I tested with it anyway just to be sure. The issue persists with 16.2.5.

#10 - 08/10/2021 01:42 PM - Andrew Davidoff

Based on the progress here it seems like I'm probably the only person to have reported this. I still can't figure out why that'd be. I wonder if you have had a chance to look at the core dump and/or reproduce this and if you have an idea of what's going on here? It may help me mitigate on my end if nothing else. Thanks.

#11 - 08/19/2021 08:45 PM - Neha Ojha

- Assignee set to Adam Kupczyk

Adam, can you start talking a look at this?

#12 - 09/02/2021 05:48 PM - Andrew Davidoff

I dug into this more today and I am wondering if it has something to do with `_conf->cluster` not being set right (to the default of `ceph`). Unfortunately editing the OSD's `unit.run` to include `--cluster ceph` in the arg list didn't change the behavior, so no additional clue provided there.

#13 - 11/09/2021 06:21 PM - André Cruz

I'm also seeing this issue:

```
*** Caught signal (Segmentation fault) **
 in thread 7f6a7e69c700 thread_name:ceph-mon
 ceph version 16.2.6 (ee28fb57e47e9f88813e24bbf4c14496ca299d31) pacific (stable)
 1: /lib64/libpthread.so.0(+0x12b20) [0x7f6a73562b20]
 2: /lib64/libc.so.6(+0x15d5f5) [0x7f6a722e95f5]
 3: (SubProcess::add_cmd_arg(char const*)+0x4c) [0x7f6a75a5cbfc]
 4: (SubProcess::add_cmd_args(char const*, ...)+0x75) [0x7f6a75a5cd95]
 5: (ceph::crush::CrushLocation::update_from_hook()+0x2d4) [0x7f6a75de45e4]
 6: (ceph::crush::CrushLocation::init_on_startup()+0x385) [0x7f6a75de56c5]
 7: (global_init(std::map<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std
::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::less<std::__cxx11::basic_st
ring<char, std::char_traits<char>, std::alloca
tor<char> > >, std::allocator<std::pair<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocato
r<char> > const, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > > > > const*,
 std::vector<char const*, std::allocator<char
const*> >&, unsigned int, code_environment_t, int, bool)+0xcd1) [0x55eadd92b941]
 8: main()
 9: __libc_start_main()
 10: _start()
```

And I also use a bash script as crush hook. The script works fine, and is used in Nautilus clusters with no issue:

```
$ /usr/local/bin/cf-crush-hook
root=default datacenter=dc1 rack=A2 host=hostxxx
```

#14 - 11/16/2021 01:24 PM - Adam Kupczyk

- Backport set to pacific

- Pull request ID set to 43944

#15 - 11/16/2021 03:12 PM - Andrew Davidoff

Thank you for this fix. It is very much appreciated.

#16 - 11/16/2021 10:15 PM - Neha Ojha

- Status changed from New to Fix Under Review

#17 - 12/02/2021 11:33 PM - Neha Ojha

- Status changed from Fix Under Review to Pending Backport

#18 - 12/02/2021 11:36 PM - Backport Bot

- Copied to Backport #53480: pacific: Segmentation fault under Pacific 16.2.1 when using a custom crush location hook added

#19 - 01/12/2022 10:03 AM - Janek Bevendorff

This present in 16.2.7. Any reason why the linked PR wasn't merged into that release?

#20 - 02/22/2022 04:11 PM - Chris Durham

This issue bit us in our upgrade to 16.2.7 from 15.2.15. We have a manual cluster (non-cephadm). We followed the procedure at:

https://docs.ceph.com/en/latest/releases/pacific/ under "Upgrading non-cephadm clusters"

We have several mixed servers (hdd and ssd), and used the hook functionality to segregate the ssds into their own virtual 'hosts', and then created rules that two pools were using for their crush rule. When we discovered the SIGSEGV after upgrading the mons, the data in the dump led us to this bug. Luckily, after removing the crush hook and getting the mons to start, the osds had not yet been restarted, and were still using the old crush map that had some osds remapped to their own virtual 'hosts'. We were able to change the crush rule of the pools in question before restarting the OSDs, and as such did not lose data.

https://docs.ceph.com/en/latest/releases/pacific/
https://ceph.io/en/news/blog/2021/v16-2-7-pacific-released/
https://docs.ceph.com/en/latest/cephadm/upgrade/

Nowhere that I can find in any of the above is this bug mentioned. This bug MUST be mentioned in the upgrade process, "Don't upgrade if you rely on crush hooks and cannot remap" just like the "bluestore-quick-fix-on-mount" issue is mentioned for a 16.2.6 upgrade.

#21 - 02/22/2022 05:09 PM - Chris Durham

Chris Durham wrote:

> This issue bit us in our upgrade to 16.2.7 from 15.2.15. We have a manual cluster (non-cephadm). We followed the procedure at:
>
> https://docs.ceph.com/en/latest/releases/pacific/ under "Upgrading non-cephadm clusters"
>
> We have several mixed servers (hdd and ssd), and used the hook functionality to segregate the ssds into their own virtual 'hosts', and then created rules that two pools were using for their crush rule. When we discovered the SIGSEGV after upgrading the mons, the data in the dump led us to this bug. Luckily, after removing the crush hook and getting the mons to start, the osds had not yet been restarted, and were still using the old crush map that had some osds remapped to their own virtual 'hosts'. We were able to change the crush rule of the pools in question before restarting the OSDs, and as such did not lose data.
>
> https://docs.ceph.com/en/latest/releases/pacific/
> https://ceph.io/en/news/blog/2021/v16-2-7-pacific-released/
> https://docs.ceph.com/en/latest/cephadm/upgrade/
>
> Nowhere that I can find in any of the above is this bug mentioned. This bug MUST be mentioned in the upgrade process, "Don't upgrade if you rely on crush hooks and cannot remap" just like the "bluestore-quick-fix-on-mount" issue is mentioned for a 16.2.6 upgrade.

Note this was on CentOS 8.5

#22 - 03/03/2022 03:09 PM - Wyllys Ingersoll

This seems to be a pretty high priority issue, we just hit it upgrading from nautilus to 16.2.7 on a cluster with 100+ osds in various configurations. Is it going to be fixed in the next Pacific update?

#23 - 04/06/2022 12:11 AM - Andrew Davidoff

I appreciate the work to get this bug squashed but I wonder if there's a schedule published somewhere that might indicate when it could be merged into another Pacific release?

#24 - 04/06/2022 12:41 AM - Neha Ojha

Andrew Davidoff wrote:

> I appreciate the work to get this bug squashed but I wonder if there's a schedule published somewhere that might indicate when it could be merged into another Pacific release?

We are planning a 16.2.8 release in the next few weeks, which will include this fix.

#25 - 05/31/2022 02:41 PM - Neha Ojha

- Status changed from Pending Backport to Resolved

Files

osd-segfault-when-crush-location-hook-configured.log	66.6 KB	05/05/2021	Andrew Davidoff
core.ceph-osd.1620430233.gz	936 KB	05/07/2021	Andrew Davidoff