RADOS - Bug #39174

crushtool crash on Fedora 28 and newer

04/10/2019 08:30 PM - Ken Dreyer

Status:	Resolved	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Brad Hubbard		
Category:			
Target version:			
Source:	Q/A	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	nautilus, mimic, luminous	**Component(RADOS):**	
Regression:	No	**Pull request ID:**	27506
Severity:	3 - minor	**Crash signature (v1):**	
Reviewed:		**Crash signature (v2):**	

Description

On Fedora 29, Fedora 30, and RHEL 8, /usr/bin/crushtool crashes when trying to compile the map that Rook uses.

```
#0  0x00007fffeecf053f in raise () from /lib64/libc.so.6
#1  0x00007fffeecda895 in abort () from /lib64/libc.so.6
#2  0x00007fffef71e7a8 in std::__replacement_assert(char const*, int, char const*, char const*) ()
 from /usr/lib64/ceph/libceph-common.so.0
#3  0x00007fffef93a063 in std::vector<int, std::allocator<int> >::operator[](unsigned long) () fro
m /usr/lib64/ceph/libceph-common.so.0
#4  0x00007fffefb882a5 in CrushCompiler::parse_bucket(__gnu_cxx::__normal_iterator<boost::spirit::
tree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> >*, std::vector<boost::s
pirit::tree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> >, std::allocator
<boost::spirit::tree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> > > > >
const&)
    () from /usr/lib64/ceph/libceph-common.so.0
#5  0x00007fffefb88ab0 in CrushCompiler::parse_crush(__gnu_cxx::__normal_iterator<boost::spirit::t
ree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> >*, std::vector<boost::sp
irit::tree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> >, std::allocator<
boost::spirit::tree_node<boost::spirit::node_val_data<char const*, boost::spirit::nil_t> > > > > c
onst&)
    () from /usr/lib64/ceph/libceph-common.so.0
#6  0x00007fffefb8aee8 in CrushCompiler::compile(std::istream&, char const*) ()
   from /usr/lib64/ceph/libceph-common.so.0
#7  0x0000555555562e13 in main (argc=<optimized out>, argv=<optimized out>)
    at /usr/include/c++/8/bits/basic_string.h:2290
```

The crushmap.txt is:

```
# begin crush map
tunable choose_local_tries 0
tunable choose_local_fallback_tries 0
tunable choose_total_tries 50
tunable chooseleaf_descend_once 1
tunable chooseleaf_vary_r 1
tunable chooseleaf_stable 0
tunable straw_calc_version 1
tunable allowed_bucket_algs 22

# types
```

```
type 0 osd
type 1 host
type 2 chassis
type 3 rack
type 4 row
type 5 pdu
type 6 pod
type 7 room
type 8 datacenter
type 9 region
type 10 root

# default bucket
root default {
    id -1   # do not change unnecessarily
    alg straw
    hash 0  # rjenkins1
}

# rules
rule replicated_ruleset {
    ruleset 0
    type replicated
    min_size 1
    max_size 10
    step take default
    step chooseleaf firstn 0 type host
    step emit
}

# end crush map
```

This crash occurs on the following platforms:

- ceph-base-12.2.7-1.fc28
- ceph-base-12.2.11-1.fc28
- ceph-base-12.2.11-1.fc29
- ceph-base-14.2.0-1.fc30
- tip of nautilus on RHEL 8

It does not crash on:

- ceph-base-12.2.8-1.fc27

One difference I see between Fedora 27 and 28 is that Fedora 27 has libstdc++-7.3.1 and Fedora 28 has libstdc++-8.3.1 , but that is just a guess.

Related issues:

Copied to RADOS - Backport #39309: luminous: crushtool crash on Fedora 28 and...	**Rejected**
Copied to RADOS - Backport #39310: nautilus: crushtool crash on Fedora 28 and...	**Resolved**
Copied to RADOS - Backport #39311: mimic: crushtool crash on Fedora 28 and newer	**Resolved**

History

#1 - 04/10/2019 08:31 PM - Ken Dreyer

- Description updated

#2 - 04/10/2019 08:37 PM - Ken Dreyer

- Subject changed from crushmap crash on Fedora 29 and newer to crushmap crash on Fedora 28 and newer

- Description updated

#3 - 04/10/2019 08:39 PM - Ken Dreyer

- Subject changed from crushmap crash on Fedora 28 and newer to crushtool crash on Fedora 28 and newer

- Description updated

#4 - 04/10/2019 08:49 PM - Ken Dreyer

- Description updated

#5 - 04/10/2019 08:52 PM - Ken Dreyer

- Priority changed from Normal to Urgent

#6 - 04/10/2019 09:30 PM - Vasu Kulkarni

very good reason to drop one distro in teuthology and replace it with fedora 28, I think Brad brought this up long time back too in #sepia.

#7 - 04/11/2019 02:30 AM - Brad Hubbard

Vasu Kulkarni wrote:

> very good reason to drop one distro in teuthology and replace it with fedora 28, I think Brad brought this up long time back too in #sepia.

Many times, long ago, yes.

I'm looking into this crash.

#8 - 04/11/2019 02:31 AM - Brad Hubbard

- Project changed from mgr to RADOS

- Status changed from New to 12

- Assignee set to Brad Hubbard

- Source set to Q/A

#9 - 04/11/2019 04:36 AM - Brad Hubbard

Turning up verbosity gives clues to what might be the problem.

```
<mock-chroot> sh-4.4# ./crushtool -v -c crushmap.txt 2>&1|head -25
tunable choose_local_tries 0
tunable choose_local_fallback_tries 0
tunable choose_total_tries 50
tunable chooseleaf_descend_once 1
tunable chooseleaf_vary_r 1
tunable chooseleaf_stable 0
tunable straw_calc_version 1
tunable allowed_bucket_algs 22
type 0 'osd'
type 1 'host'
type 2 'chassis'
type 3 'rack'
type 4 'row'
type 5 'pdu'
type 6 'pod'
type 7 'room'
type 8 'datacenter'
type 9 'region'
type 10 'root'
```

```
bucket default id -1
bucket default (-1) 0 items and weight 0
/usr/include/c++/8/bits/stl_vector.h:932: std::vector<_Tp, _Alloc>::reference std::vector<_Tp, _Alloc>::operat
or[](std::vector<_Tp, _Alloc>::size_type) [with _Tp = int; _Alloc = std::allocator<int>; std::vector<_Tp, _All
oc>::reference = int&; std::vector<_Tp, _Alloc>::size_type = long unsigned int]: Assertion '__builtin_expect(_
_n < this->size(), true)' failed.
*** Caught signal (Aborted) **
 in thread 7f64629f6540 thread_name:crushtool
 ceph version 12.2.11 (26dc3775efc7bb286a1d6d66faee0ba30ea23eee) luminous (stable)
```

The problem here is we have the following code in src/crush/CrushCompiler.cc

```
 561 int CrushCompiler::parse_bucket(iter_t const& i)

 562 {
...
 651   vector<int> items(size);

 652   vector<int> weights(size);
...
 746   int r = crush.add_bucket(id, alg, hash, type, size,

 747                            &items[0], &weights[0], &idout);
```

Looking at a coredump.

```
(gdb) f
#4  0x00007fffefb86e85 in CrushCompiler::parse_bucket (this=0x7fffffffcfe0, i=...) at /builddir/build/BUILD/ce
ph-12.2.11/src/crush/CrushCompiler.cc:746
746       int r = crush.add_bucket(id, alg, hash, type, size,
(gdb) l
741       item_id[name] = id;
742       item_weight[id] = bucketweight;
743
744       assert(id != 0);
745       int idout;
746       int r = crush.add_bucket(id, alg, hash, type, size,
747                                &items[0], &weights[0], &idout);
748       if (r < 0) {
749         if (r == -EEXIST)
750           err << "Duplicate bucket id " << id << std::endl;
(gdb) p items
$1 = std::vector of length 0, capacity 0
(gdb) p weights
$2 = std::vector of length 0, capacity 0
(gdb) down
#3  0x00007fffef936783 in std::vector<int, std::allocator<int> >::operator[] (this=this@entry=0x7fffffffc5a0,
__n=__n@entry=0) at /usr/include/c++/8/bits/stl_vector.h:805
805           size() const _GLIBCXX_NOEXCEPT
(gdb)
#2  0x00007fffef716168 in std::__replacement_assert (__file=__file@entry=0x7fffefc134c0 "/usr/include/c++/8/bi
ts/stl_vector.h", __line=__line@entry=932,
    __function=__function@entry=0x7fffefc47ca0 <_ZZNSt6vectorIiSaIiEEixEmE19__PRETTY_FUNCTION__> "std::vector<
_Tp, _Alloc>::reference std::vector<_Tp, _Alloc>::operator[](std::vector<_Tp, _Alloc>::size_type) [with _Tp =
int; _Alloc = std::allocator<int>; std::vector<_Tp, _Alloc>::reference = int&;"..., __condition=__condition@en
try=0x7fffefc13490 "__builtin_expect(__n < this->size(), true)") at /usr/include/c++/8/x86_64-redhat-linux/bit
s/c++config.h:2391
2391        __builtin_abort();
(gdb) l
2386      __replacement_assert(const char* __file, int __line,
2387                           const char* __function, const char* __condition)
2388      {
2389        __builtin_printf("%s:%d: %s: Assertion '%s' failed.\n", __file, __line,
2390                         __function, __condition);
2391        __builtin_abort();
2392      }
```

```
2393    }
2394    #define __glibcxx_assert_impl(_Condition)                                    \
2395      do
(gdb) printf "Assertion '%s' failed.\n", __condition
Assertion '__builtin_expect(__n < this->size(), true)' failed.
(gdb) up
#3  0x00007fffef936783 in std::vector<int, std::allocator<int> >::operator[] (this=this@entry=0x7fffffffc5a0,
__n=__n@entry=0) at /usr/include/c++/8/bits/stl_vector.h:805
805           size() const _GLIBCXX_NOEXCEPT
(gdb) p __n
$3 = 0
(gdb) p  this->size()
$4 = 0
```

Well fair enough. So why here? Why now?

Due to the inclusion of _GLIBCXX_ASSERTIONS in the CXXFLAGS. The use of the address of element 0 of an empty vector is considered unsafe although it will historically do what you want. However, here we are being pulled up on it. I suspect we need to pass the data() [1] member function of vector here but I'll need to do some testing.

[1] http://www.open-std.org/jtc1/sc22/wg21/docs/lwg-defects.html#464 first line under "Rationale:"

#10 - 04/11/2019 06:02 AM - Brad Hubbard

https://bugzilla.redhat.com/show_bug.cgi?id=1515858

#11 - 04/11/2019 07:10 AM - Brad Hubbard

- Status changed from 12 to Fix Under Review

#12 - 04/11/2019 07:11 AM - Brad Hubbard

- Pull request ID set to 27506

#13 - 04/11/2019 07:56 AM - Brad Hubbard

- Status changed from Fix Under Review to In Progress

#14 - 04/11/2019 08:24 AM - Brad Hubbard

- Backport set to nautilus, mimic, luminous

#15 - 04/14/2019 08:26 PM - Sage Weil

- Status changed from In Progress to Pending Backport

#16 - 04/16/2019 08:00 AM - Nathan Cutler

- Copied to Backport #39309: luminous: crushtool crash on Fedora 28 and newer added

#17 - 04/16/2019 08:00 AM - Nathan Cutler

- Copied to Backport #39310: nautilus: crushtool crash on Fedora 28 and newer added

#18 - 04/16/2019 08:00 AM - Nathan Cutler

- Copied to Backport #39311: mimic: crushtool crash on Fedora 28 and newer added

#19 - 01/27/2021 07:12 PM - Nathan Cutler

- 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".