# Ceph - Bug #16653

## ceph mon Segmentation fault after set crush_ruleset ceph 10.2.2

07/11/2016 04:14 PM - Oliver Dzombc

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **% Done:** | 0% |
| **Priority:** | Normal | **Spent time:** | 0.00 hour |
| **Assignee:** | Xiaoxi Chen | | |
| **Category:** | | | |
| **Target version:** | | | |
| **Source:** | | **Reviewed:** | |
| **Tags:** | | **Affected Versions:** | |
| **Backport:** | jewel | **ceph-qa-suite:** | |
| **Regression:** | | **Pull request ID:** | |
| **Severity:** | | **Crash signature:** | |

**Description**

Hi,

1. ceph osd pool create lxc 128
2. ceph osd pool set lxc crush_ruleset 2

cause mon's to be killed:

http://pastebin.com/rv2yPpjZ

Aborting to set the crush_ruleset will show

http://pastebin.com/qm7Ydbd6

---

While the output at the mon looks like:

http://pastebin.com/D1UUfLFK

1. ceph osd pool ls detail

pool 3 'ssd_cache' replicated size 2 min_size 1 crush_ruleset 1
object_hash rjenkins pg_num 1024 pgp_num 1024 last_change 237 flags
hashpspool,incomplete_clones tier_of 4 cache_mode writeback target_bytes
850000000000 hit_set bloom{false_positive_probability: 0.05,
target_size: 0, seed: 0} 120s x1 decay_rate 0 search_last_n 0 stripe_width 0

pool 4 'cephfs_data' replicated size 2 min_size 1 crush_ruleset 2
object_hash rjenkins pg_num 1024 pgp_num 1024 last_change 169 lfor 144
flags hashpspool crash_replay_interval 45 tiers 3 read_tier 3 write_tier
3 stripe_width 0

pool 5 'cephfs_metadata' replicated size 2 min_size 1 crush_ruleset 1
object_hash rjenkins pg_num 128 pgp_num 128 last_change 191 flags
hashpspool stripe_width 0

pool 7 'lxc' replicated size 2 min_size 1 crush_ruleset 1 object_hash
rjenkins pg_num 128 pgp_num 128 last_change 473 flags hashpspool
stripe_width 0

This here is from the mon server which issues the command:

http://pastebin.com/b2bCJsGT

OS is Centos 7, default kernel.

Any idea what the problem is ? Cluster is healthy, same command could be issued successfully in the past, world seems fine.

Thank you !

Greetings
Oliver

**Related issues:**

| | | |
|---|---|---|
| Duplicated by Ceph - Bug #17412: Applying ruleset halts monitor | **Duplicate** | **09/27/2016** |
| Copied to Ceph - Backport #17135: jewel: ceph mon Segmentation fault after se... | **Resolved** | |

## History

#### #1 - 07/12/2016 06:51 AM - Xiaoxi Chen

Tried but didnt reproduce.
Did you stably reproduce it?

#### #2 - 07/12/2016 07:41 AM - Oliver Dzombc

Hi,

jep, happens every time, 100% "success".

#### #3 - 07/12/2016 08:52 AM - Oliver Dzombc

Here is the current crushmap:

1. 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 straw_calc_version 1

1. devices
   device 0 osd.0
   device 1 osd.1
   device 2 osd.2
   device 3 osd.3
   device 4 osd.4
   device 5 osd.5
   device 6 osd.6
   device 7 osd.7
   device 8 osd.8
   device 9 osd.9
   device 10 osd.10
   device 11 osd.11
   device 12 osd.12
   device 13 osd.13
   device 14 osd.14
   device 15 osd.15

1. 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

1. buckets
   host cephosd2-ssd-cache {

```
id -1           # do not change unnecessarily        # weight 0.872
alg straw
hash 0  # rjenkins1
item osd.8 weight 0.218
item osd.9 weight 0.218
item osd.10 weight 0.218
item osd.11 weight 0.218
}
host cephosd2-cold-storage {
id -2           # do not change unnecessarily        # weight 14.548
alg straw
hash 0  # rjenkins1
item osd.12 weight 3.637
item osd.13 weight 3.637
item osd.14 weight 3.637
item osd.15 weight 3.637
}
host cephosd1-ssd-cache {
id -3           # do not change unnecessarily        # weight 0.872
alg straw
hash 0  # rjenkins1
item osd.0 weight 0.218
item osd.1 weight 0.218
item osd.2 weight 0.218
item osd.3 weight 0.218
}
host cephosd1-cold-storage {
id -4           # do not change unnecessarily        # weight 14.548
alg straw
hash 0  # rjenkins1
item osd.4 weight 3.637
item osd.5 weight 3.637
item osd.6 weight 3.637
item osd.7 weight 3.637
}
root ssd-cache {
id -5           # do not change unnecessarily        # weight 1.704
alg straw
hash 0  # rjenkins1
item cephosd1-ssd-cache weight 0.852
item cephosd2-ssd-cache weight 0.852
}
root cold-storage {
id -6           # do not change unnecessarily        # weight 29.094
alg straw
hash 0  # rjenkins1
item cephosd1-cold-storage weight 14.547
item cephosd2-cold-storage weight 14.547
}
```

1. rules

```
rule ssd-cache-rule {
ruleset 1
type replicated
min_size 2
max_size 10
step take ssd-cache
step chooseleaf firstn 0 type host
step emit
}
rule cold-storage-rule {
ruleset 2
type replicated
min_size 2
max_size 10
step take cold-storage
step chooseleaf firstn 0 type host
step emit
}
```

1. end crush map

#### #4 - 07/13/2016 07:23 AM - Oliver Dzombc

If i set:

1. ceph osd pool create vmware1 64 cold-storage-rule
   pool 'vmware1' created

I would expect the pool to have ruleset 2.

#ceph osd pool ls detail

pool 10 'vmware1' replicated size 3 min_size 2 crush_ruleset 1
object_hash rjenkins pg_num 64 pgp_num 64 last_change 483 flags
hashpspool stripe_width 0

but it has crush_ruleset 1.

#### #5 - 07/14/2016 01:14 PM - Oliver Dzombc

Hi,

so is there anything i can do, to get more info about it ?

Its a big problem, that we can not add any pools. crush_ruleset 1 is the ssd cache tier, so holding pool data in there, is somehow not really wanted.

Thank you !

#### #6 - 07/14/2016 10:40 PM - Oliver Dzombc

Hi Xiaoxi Chen,

that you have something to reproduce:

Edit your crushmap, remove ruleset 0.

So if your crushmap does not have a ruleset 0, you have the bug.

My crushmap had ruleset 1 and 2. There was no 0.

That cause the bug, reproduceable. After i fixed it, its working again as expected.

#### #7 - 07/15/2016 11:44 AM - Artemy Kapitula

Exactly the same problem on 10.2.1.

```
ceph version 10.2.1 (3a66dd4f30852819c1bdaa8ec23c795d4ad77269)
 1: (()+0x5054ba) [0x5626fe81a4ba]
 2: (()+0xf100) [0x7f5e7446f100]
 3: (OSDMonitor::prepare_command_pool_set(std::map&lt;std::string, boost::variant&lt;std::string, bool, long,
double, std::vector&lt;std::string, std::allocator&lt;std::string&gt; >, boost::detail::variant::void_, boost
::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant
::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant:
:void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::de
tail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::det
ail::variant::void_>, std::less&lt;std::string&gt;, std::allocator&lt;std::pair&lt;std::string const, boost::v
ariant&lt;std::string, bool, long, double, std::vector&lt;std::string, std::allocator&lt;std::string&gt; >, bo
ost::det
ail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::voi
d_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void
, boost::detail::variant::void
, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::
variant::void_, boost::detail::variant::void_, boost::detail::
variant::void_, boost::detail::variant::void_> > > >&, std::basic_stringstream&lt;char, std::char_traits&lt;ch
ar&gt;, std::allocator&lt;char&gt; >&)+0x122f) [0x5626fe6268df]
 4: (OSDMonitor::prepare_command_impl(std::shared_ptr&lt;MonOpRequest&gt;, std::map&lt;std::string, boost::var
iant&lt;std::string, bool, long, double, std::vector&lt;std::string, std::allocator&lt;std::string&gt; >, boos
t::de
tail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::vo
id_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::voi
d_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail
::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail:
:variant::void_, boost::detail::variant::void_>, std::less&lt;std::string&gt;, std::allocator&lt;std::pair&lt;
std::string const, boost::variant&lt;std::string, bool, long, double, std::vector&lt;std::string, std::allocat
or
&lt;std::string&gt; >, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::v
oid_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, b
oost::detail::variant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::var
iant::void_, boost::detail::variant::void_, boost::detail::variant::void_, boost::detail::vari
ant::void_, boost::detail::variant::void_, boost::detail::variant::void_> > > >&)+0xf02c) [0x5626fe6365ec]
 5: (OSDMonitor::prepare_command(std::shared_ptr&lt;MonOpRequest&gt;)+0x64f) [0x5626fe63b3cf]
 6: (OSDMonitor::prepare_update(std::shared_ptr&lt;MonOpRequest&gt;)+0x307) [0x5626fe63cf27]
 7: (PaxosService::dispatch(std::shared_ptr&lt;MonOpRequest&gt;)+0xe0b) [0x5626fe5eb51b]
 8: (Monitor::handle_command(std::shared_ptr&lt;MonOpRequest&gt;)+0x1d1f) [0x5626fe5a753f]
 9: (Monitor::dispatch_op(std::shared_ptr&lt;MonOpRequest&gt;)+0x33b) [0x5626fe5b30bb]
 10: (Monitor::_ms_dispatch(Message*)+0x6c9) [0x5626fe5b4459]
 11: (Monitor::handle_forward(std::shared_ptr&lt;MonOpRequest&gt;)+0x89c) [0x5626fe5b28ec]
 12: (Monitor::dispatch_op(std::shared_ptr&lt;MonOpRequest&gt;)+0xc70) [0x5626fe5b39f0]
 13: (Monitor::_ms_dispatch(Message*)+0x6c9) [0x5626fe5b4459]
 14: (Monitor::ms_dispatch(Message*)+0x23) [0x5626fe5d4f73]
 15: (DispatchQueue::entry()+0x78a) [0x5626fea2d9fa]
 16: (DispatchQueue::DispatchThread::entry()+0xd) [0x5626fe92310d]
 17: (()+0x7dc5) [0x7f5e74467dc5]
 18: (clone()+0x6d) [0x7f5e72d3028d]
 NOTE: a copy of the executable, or `objdump -rdS &lt;executable&gt;` is needed to interpret this.
```

#### #8 - 07/15/2016 11:59 AM - Oliver Dzombc

Hi Artemy,

did you already check my work around ?

Simply add a ruleset with id 0 and default.

Something like:

rule default {
ruleset 0
type replicated
min_size 2
max_size 10
step chooseleaf firstn 0 type host
step emit
}

Should already fix the effect of the issue.

#### #9 - 07/16/2016 05:43 AM - Xiaoxi Chen

Hi Oliver Dzombc,
would you mind paste the PR link here?

#### #10 - 07/18/2016 07:24 AM - Artemy Kapitula

> did you already check my work around ?
> Simply add a ruleset with id 0 and default.

Hi Oliver!

Yes, I tried today on test/dev cluster.
No effect.
2 of 3 mons crashed.

But we've got 10.2.1 now, not 10.2.2.

#### #11 - 07/18/2016 07:53 AM - Oliver Dzombc

Hi,

if you created >exactly<

rule default {
ruleset 0
type replicated
min_size 2
max_size 10
step chooseleaf firstn 0 type host
step emit
}

as rule, then no idea.

If not, please create exactly that rule and try it out.

Good Luck !

#### #12 - 07/18/2016 08:42 AM - Xiaoxi Chen

*- Assignee set to Xiaoxi Chen*

#### #13 - 07/18/2016 04:36 PM - Xiaoxi Chen

Likely fixed by this commit https://github.com/ceph/ceph/pull/8480

The problem is in 10.2.2 code we assume ruleset N is located in crush->rules[N], but this is not always true. In your case, because you don't have ruleset 0, so when importing, ruleset 1 is in rules[0] while ruleset 2 is in rules[1]. Then when you set the ruleset of one pool to 2, in osdmap.crush->get_rule_mask_min_size(n), it will access rules[2] , definitely get a Segmentation fault.

Use "crush rule rm" to delete ruleset will not hit this bug, because the command just set crush->rules[N] to NULL instead of re-placing them.

@Artemy Kapitula, @Oliver Dzombc. It would be great if you could test against master (or cherry-pick this commit ), and maybe we would need to backport this.

#### #14 - 07/25/2016 06:16 AM - Artemy Kapitula

Hi Xiaoxi Chen!

I did a test with special conditions: three rulesets with ids=0,2,3:

rule replicated_ruleset {
ruleset 0
type replicated
min_size 1

```
max_size 10
step take default
step choose firstn 0 type osd
step emit
}

rule bbb {
ruleset 2
type replicated
min_size 1
max_size 10
step take default
step chooseleaf firstn 0 type osd
step emit
}

rule aaa {
ruleset 3
type replicated
min_size 1
max_size 10
step take default
step chooseleaf firstn 0 type osd
step emit
}
```

set crush_ruleset works fine with rulesets=0,2, but breaks in segfault with ruleset=3.
The only workaround I found is to keep all rulesets up to max(id) existing.
But after a rule removal it all may crash down on the first set crush_ruleset :-)
I'll try to build ceph with patches suggested, but that will take some time.

#### #15 - 07/26/2016 06:03 AM - Artemy Kapitula

Xiaoxi Chen wrote:

> Likely fixed by this commit https://github.com/ceph/ceph/pull/8480

Confirmed, set crush_ruleset now works well.

#### #16 - 07/27/2016 03:36 PM - Nathan Cutler

*- Target version deleted (519)*

#### #17 - 08/23/2016 04:16 PM - Kefu Chai

might need to backport https://github.com/ceph/ceph/pull/8480 to jewel

#### #18 - 08/23/2016 04:17 PM - Kefu Chai

*- Tracker changed from Bug to Backport*

#### #19 - 08/25/2016 11:45 AM - Loic Dachary

*- Tracker changed from Backport to Bug*

*- Status changed from New to Pending Backport*

*- % Done set to 0*

*- Backport set to jewel*

#### #20 - 08/25/2016 11:46 AM - Loic Dachary

*- Copied to Backport #17135: jewel: ceph mon Segmentation fault after set crush_ruleset ceph 10.2.2 added*

#### #21 - 10/14/2016 05:03 PM - Nathan Cutler

*- Status changed from Pending Backport to Resolved*

#### #22 - 05/24/2017 03:40 PM - Sage Weil

*- Duplicated by Bug #17412: Applying ruleset halts monitor added*