

CephFS - Bug #17466

MDSMonitor: non-existent standby_for_fscid not caught

09/30/2016 08:48 PM - Greg Farnum

Status: Resolved	% Done: 0%
Priority: Urgent	
Assignee: John Spray	
Category: Correctness/Safety	
Target version:	
Source: Community (user)	Affected Versions:
Tags:	ceph-qa-suite:
Backport: jewel	Component(FS): MDSMonitor
Regression: No	Labels (FS):
Severity: 3 - minor	Pull request ID:
Reviewed:	Crash signature:
Description	
<p>We've got it using aggregate initialization (that is, the explicitity {a, b} syntax), but we're ordering the rank and fscid the way they are in the constructor — which is backwards from how they are declared in the struct. So it's swapped and if anything hits this code path it goes dramatically wrong.</p> <p>I pushed a quick fix to wip-fixup-mds-standby-init that should resolve it by swapping them into the right sequence, but we probably want to make the struct declaration match the constructor order instead. Adam Tygart/m0zes__ ran into this bug and will be testing my fix.</p>	
Related issues:	
Copied to CephFS - Backport #17557: jewel: MDSMonitor: non-existent standby_f...	Resolved

History

#1 - 09/30/2016 08:59 PM - John Spray

- Status changed from New to Fix Under Review

<https://github.com/ceph/ceph/pull/11281>

#2 - 09/30/2016 09:50 PM - John Spray

So now it seems like although the order of members was whacky, calls to = {fscid, rank} were actually assigning the values as expected.

The other possibilities would be that legacy_client_fscid was something invalid (but it is validated in FSMap::sanity), or standby_for_fscid was something non-default (m0zes says it wasn't).

#3 - 10/03/2016 06:30 AM - John Spray

- Subject changed from MDSMonitor: maybe_promote_standby is initializing mds_role_t wrong to MDSMonitor: non-existent standby_for_fscid not caught

```
(gdb) print info
$7 = (const MDSMap::mds_info_t &) @0x555555fb1da68: {
  global_id = {<boost::totally_ordered1<mds_gid_t,
boost::totally_ordered2<mds_gid_t, unsigned long,
boost::detail::empty_base<mds_gid_t> > >> =
{<boost::less_than_comparable1<mds_gid_t,
boost::equality_comparable1<mds_gid_t,
boost::totally_ordered2<mds_gid_t, unsigned long,
```

```

boost::detail::empty_base<mds_gid_t> > > > =
{<boost::equality_comparable1<mds_gid_t,
boost::totally_ordered2<mds_gid_t, unsigned long,
boost::detail::empty_base<mds_gid_t> > > > =
{<boost::totally_ordered2<mds_gid_t, unsigned long,
boost::detail::empty_base<mds_gid_t> > > > =
{<boost::less_than_comparable2<mds_gid_t, unsigned long,
boost::equality_comparable2<mds_gid_t, unsigned long,
boost::detail::empty_base<mds_gid_t> > > > =
{<boost::equality_comparable2<mds_gid_t, unsigned long,
boost::detail::empty_base<mds_gid_t> > > > =
{<boost::detail::empty_base<mds_gid_t>> = {<No data fields>}, <No data
fields>}, <No data fields>}, <No data fields>}, <No data fields>}, <No
data fields>}, <No data fields>}, t = 1055992652}, name = "mormo",
rank = -1, inc = 0,
  state = MDSMap::STATE_STANDBY, state_seq = 2, addr = {type = 0,
nonce = 8835, {addr = {ss_family = 2, __ss_align = 0, __ss_padding =
'\000' <repeats 111 times>}, addr4 = {sin_family = 2, sin_port =
36890,
  sin_addr = {s_addr = 50398474}, sin_zero =
"\000\000\000\000\000\000\000"}, addr6 = {sin6_family = 2, sin6_port =
36890, sin6_flowinfo = 50398474, sin6_addr = {__in6_u = {
  __u6_addr8 = '\000' <repeats 15 times>, __u6_addr16 = {0,
0, 0, 0, 0, 0, 0, 0}, __u6_addr32 = {0, 0, 0, 0}}, sin6_scope_id =
0}}}, laggy_since = {tv = {tv_sec = 0, tv_nsec = 0}},
  standby_for_rank = 0, standby_for_name = "", standby_for_fscid =
328, standby_replay = true, export_targets = std::set with 0 elements,
mds_features = 1967095022025}
(gdb) print target_role
$8 = {rank = 0, fscid = <optimized out>}

```

#4 - 10/03/2016 06:39 AM - John Spray

I think this is happening when older MDSs send a beacon to newer mons, because MMDSBeacon isn't initialising POD fields properly in its default constructor, so when there's no ::decode for fields like standby_for_fscid we're getting garbage.

#5 - 10/03/2016 06:46 AM - John Spray

wip-17466-jewel is building with a minimal fix rebased on v10.2.3

<https://github.com/ceph/ceph/pull/11281> has the fix for master.

It's tempting to validate `standby_for_fscid` in `FSMap::sanity()`, but we don't do that for `standby_for_rank` because daemons are allowed to have config that references a rank that perhaps doesn't actually exist in the map. It's up to the logic that consumes `standby_for_*` to avoid trusting what has been put there by the daemon.

#6 - 10/03/2016 03:08 PM - Adam Tygart

I've cherry-picked the mds and mon fixes from your pull-request onto jewel and built a new version. this allowed me to bring the mons back up. unfortunately, the mdsmap is incrementing, but it isn't dumping any of the new epochs. I'm rebuilding with the messages fix, (patching to change `MDS_STATE_NULL` to `CEPH_MDS_STATE_NULL`) in with the hope that a new mds map will be generated with the correct states.

```
# ceph mds dump
dumped fsmap epoch 903388
fs_name      combined
epoch       903384
flags       0
created     2016-08-08 16:42:34.345327
modified    2016-08-08 16:42:34.345327
tableserver 0
root        0
session_timeout 60
session_autoclose 300
max_file_size 17592186044416
last_failure 0
last_failure_osd_epoch 528805
compat      compat={},rocompat={},incompat={1=base v0.20,2=client writeable ranges,3=default file layouts on dirs,4=dir inode in separate object,5=mds uses versioned encoding,6=dirfrag is stored in omap,8=no anchor table}
max_mds     1
in          0
up          {}
failed      0
damaged
stopped
data_pools  100,101,102
metadata_pool 98
inline_data disabled
```

#7 - 10/03/2016 03:18 PM - Adam Tygart

- File mdsmap added

In fact, the mdsmap epoch is quite a lot larger than I remember it. I think it was in the low 50's before this issue.

Symptom of something larger?

#8 - 10/03/2016 10:06 PM - Greg Farnum

That FSMap contains a standby with "standby_for_fscid": 0', but the existing filesystem is "id": 2'. Had him set it to 2 explicitly and that did take (based on updated fsmap) but the MDS remains in standby mode. Requested a monitor log with fresh boot sequence to trace it through.

#9 - 10/03/2016 11:57 PM - Adam Tygart

ceph-post-file: 731781d4-c78c-47f9-9de3-e151b5c1a5f9

Above is the Monitor log with a fresh boot of the mds.

#10 - 10/04/2016 10:44 AM - John Spray

There is some other bug here with standby_for_rank and standby_for_fscid set correctly, I can reproduce it locally. Stand by...

#11 - 10/04/2016 11:02 AM - John Spray

Gah, I thought I saw it (created new FS and set standby_for_fscid, then saw it stuck in standby), but a few daemon restarts later I can't see it any more :-/

#12 - 10/04/2016 07:37 PM - Adam Tygart

- File fsmap-903419 added

<https://github.com/ceph/ceph/blob/master/src/mon/MDSMonitor.cc#L2865>

If I'm reading the code paths correctly, and I probably am not, the monitors try to probe to see if there are failed mds nodes. if the active one has failed, it will try to promote one from standby. After that, it reads through to promote other standby mds services to standby-replay (if enabled), following the the aforementioned promoted mds. My mds map doesn't list any failed mds services, correct? Therefore, "failed" is an empty set and nothing can be promoted to active.

I must have parsed that wrong, or else a new cephfs filesystem would never promote an mds to active, though (unless there is somewhere injecting a "failed" mds into new mdsmap?).

```
"max_mds": 1,  
"in": [  
  0  
],  
"up": {},  
"failed": [],  
"damaged": [],  
"stopped": [],
```

#13 - 10/04/2016 09:43 PM - Greg Farnum

No, you're correct. The part you're missing is `MDSMonitor::maybe_expand_cluster()`, invoked from `MDSMonitor::tick()`. I'm not seeing why that (and the `FSMap::find_replacement_for()`, `find_standby_for()`, `find_unused()` functions it calls) aren't kicking the MDS over to active.

#14 - 10/04/2016 09:57 PM - Greg Farnum

I think that if you use the repair tools to flush the MDS journal and then do an fs reset you should get everything back working again.

But I'd really like to understand how things got into this state first. Is it the result of something messed up in the point release encoding? Did you do some admin commands to your cluster that we didn't account for? The part where the MDS was insisting it was `standby_for_fscid 0` might be the root of the problem.

I've been staring at this code for a while and am at the point where really I think it just will take a bunch of debug statements added to that whole call stack to figure out where it's going wrong.

#15 - 10/04/2016 10:05 PM - Adam Tygart

there was definitely an encoding issue. I went back through the historical `mdsmaps` and it wasn't always insisting it was `standby_for_fscid 0`, it was sometimes 10000+ sometimes 0.

I haven't run any admin mds-related (or any others that I know of) commands that we haven't taken into account.

If you want to put together a branch with the debug logging you want, I'd be willing to run it tonight or tomorrow morning. I'd really like to get this bug straightened out, but I cannot wait too much longer. If a journal flush and an fs reset will help get it working I can do that tomorrow (after running the debugging branch if you want).

--

Adam

#16 - 10/04/2016 10:57 PM - Greg Farnum

I built a debug commit and merged in `wip-17466-jewel`; named the result `wip-1746-jewel-debug`.

#17 - 10/04/2016 11:34 PM - Adam Tygart

I'm building it now. What logs and levels would you like me to collect?

#18 - 10/05/2016 12:26 AM - Greg Farnum

mon level 20 should do it.

#19 - 10/05/2016 04:25 AM - Adam Tygart

<http://people.cs.ksu.edu/~mozes/ceph-mon-hobbit01-debug-branch.log>

Unfortunately, it would seem we're bailing out of `maybe_expand_cluster` before calling any of the other modified functions. Either we're down (`CEPH_MDSMAP_DOWN`), `num_in_mds >= max_mds`, or the `MDSMAP` is degraded. The middle one doesn't seem likely, the `mdsmmap` doesn't indicate (to me) that its "down" or degraded, though.

#20 - 10/05/2016 12:24 PM - John Spray

Anything rank that is in the 'in' set should also be in one of failed,stopped,damaged or up.

I notice that in the earliest logs on this issue we could see rank 0 in the 'failed' set, so something has gone wrong since.

The only way I've seen this before was if someone used rmfailed without knowing what it did?

#21 - 10/05/2016 01:06 PM - Adam Tygart

Well, crap. I don't remember running that, and it wasn't in the bash_history on the monitor I've been using, but:

```
2016-10-03 11:14:05.777419 7f6d4bbfd700 10 log_client logged 2016-10-03 11:14:04.758659 mon.0 10.5.38.1:6789/0 6961 : audit [INF] from='client.? 10.5.1.3:0/269354402' entity='client.admin' cmd='[{"prefix": "mds rmfailed", "who": "0", "confirm": "--yes-i-really-mean-it"}]': finished
2016-10-03 11:55:40.410106 7f0f213fd700 0 mon.hobbit01@0(leader) e46 handle_command mon_command({"prefix": "mds rmfailed", "who": "0", "confirm": "--yes-i-really-mean-it"} v 0) v1
2016-10-03 11:55:40.410183 7f0f213fd700 0 log_channel(audit) log [INF] : from='client.1131003972 :/0' entity='client.admin' cmd=[{"prefix": "mds rmfailed", "who": "0", "confirm": "--yes-i-really-mean-it"}]: dispatch
2016-10-03 11:55:41.056921 7f0f25b88700 0 log_channel(audit) log [INF] : from='client.1131003972 :/0' entity='client.admin' cmd='[{"prefix": "mds rmfailed", "who": "0", "confirm": "--yes-i-really-mean-it"}]': finished
```

Looks like it happened just before comment 9. Given that this portion is now probably our fault, how should I go about flushing the journal and resetting the filesystem?

#22 - 10/05/2016 01:24 PM - John Spray

- File *mds_addfailed.patch* added

Hmm, I guess the --yes-i-really-mean-it part isn't helping! (<http://tracker.ceph.com/issues/14379>)

I've attached Zheng's patch that adds an "addfailed" command (from the last time someone had this problem), so you can either recompile with that, or go ahead and do an "fs reset". My instinct is that "fs reset" should probably be okay though, as long as you stop your MDSs and clients first to avoid confusing anything too much.

#23 - 10/05/2016 02:56 PM - Adam Tygart

Digging through that patch, I'm not sure it is setup to work with the mdsmap->fsmap conversion that happened at some point in time.

Do I need to flush a journal before an fs reset?

As for the patch, I think this is what is needed to fix it up, although I'm not intimately familiar with ceph internals. As evidenced by my current issues.

```
} else if (prefix == "mds addfailed") {
-   mds_rank_t who;
```

```

-   if (!cmd_getval(g_ceph_context, cmdmap, "who", who)) {
-       ss << "error parsing 'who' value '"
-           << cmd_vartype_stringify(cmdmap["who"]) << "'";
-       return -EINVAL;
+   std::string role_str;
+   cmd_getval(g_ceph_context, cmdmap, "who", role_str);
+   mds_role_t role;
+   int rr = parse_role(role_str, &role, ss);
+   if (r < 0) {
+       ss << "invalid role '" << role_str << "'";
+       return -EINVAL;
+   }
-   pending_mdsmmap.failed.insert(who);
+   pending_fsmap.modify_filesystem(
+       role.fscid,
+       [role](std::shared_ptr<Filesystem fs)
+       {
+           fs->mds_map.failed.insert(role.rank);
+       });
+
+

```

#24 - 10/05/2016 07:43 PM - Adam Tygart

I ended up going with a journal flush and a ceph fs reset. The mds is now up and clients can mount. Thank you both for all your help, even after it became obvious that the latter half of this issue was exacerbated by me.

#25 - 10/10/2016 11:24 AM - John Spray

- Status changed from Fix Under Review to Pending Backport

Jewel backport PR here: <https://github.com/ceph/ceph/pull/11389>

#26 - 10/11/2016 08:08 AM - Loic Dachary

- Copied to Backport #17557: jewel: MDSMonitor: non-existent standby_for_fscid not caught added

#27 - 10/13/2016 03:51 PM - Loic Dachary

- Status changed from Pending Backport to Resolved

Files

mdsmmap	1022 Bytes	10/03/2016	Adam Tygart
fsmap-903419	1018 Bytes	10/04/2016	Adam Tygart
mds_addfailed.patch	2.1 KB	10/05/2016	John Spray