RADOS - Bug #37968

maybe_remove_pg_upmaps incorrectly cancels valid pending upmaps

01/18/2019 10:43 PM - Ed Fisher

Status:	Pending Backport	**Start date:**	01/18/2019
Priority:	Normal	**Due date:**	
Assignee:	xie xingguo	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:	Community (user)	**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:	luminous,mimic	**ceph-qa-suite:**	
Regression:	No	**Component(RADOS):**	
Severity:	3 - minor	**Pull request ID:**	26179

Description

It appears that OSDMap::maybe_remove_pg_upmaps's sanity checks are overzealous. With some crush rules it is possible for osdmaptool to generate valid upmaps, but maybe_remove_pg_upmaps will cancel them.

It looks like it relies on get_rule_failure_domain and rejects any upmap that results in two osds sharing a parent of that type. However, with a custom crush rule like "choose indep 2 type host, choose indep 2 type osd" such an upmap would be valid. Is it possible to use CrushWrapper::try_remap_rule or something similar to more thoroughly validate the upmap?

To reproduce:

1. ceph osd erasure-code-profile set upmaptest plugin=jerasure k=2 m=2 crush-device-class=hdd crush-failure-domain=osd
2. create a crush rule for the pool:

```
{
    "rule_id": 2,
    "rule_name": "upmaptest",
    "ruleset": 2,
    "type": 3,
    "min_size": 3,
    "max_size": 4,
    "steps": [
        {
            "op": "set_chooseleaf_tries",
            "num": 5
        },
        {
            "op": "set_choose_tries",
            "num": 100
        },
        {
            "op": "take",
            "item": -1,
            "item_name": "default"
        },
        {
            "op": "choose_indep",
            "num": 2,
            "type": "host"
        },
        {
            "op": "choose_indep",
            "num": 2,
            "type": "osd"
        },
```

```
        {
            "op": "emit"
        }
    ]
} ]
}
```

3. ceph osd pool create upmaptest 8 8 erasure upmaptest
4. Submit an upmap where the source+target osd are on the same host: ceph osd pg-upmap-items 2.7 1 2

The mon's debug log will show "2019-01-18 19:16:32.044 7fdd4d0a2700 10 maybe_remove_pg_upmaps cancel invalid pending pg_upmap_items entry 2.7->[1,2]"

This is an edge case since it depends on using a custom crush rule, but it almost completely breaks the upmap functionality for affected pools.

Related issues:

Copied to RADOS - Backport #38162: luminous: maybe_remove_pg_upmaps incorrect...	**Resolved**
Copied to RADOS - Backport #38163: mimic: maybe_remove_pg_upmaps incorrectly ...	**Need More Info**

History

#1 - 01/19/2019 06:24 AM - xie xingguo

- Assignee set to xie xingguo

#2 - 02/02/2019 09:25 AM - xie xingguo

- Status changed from New to Pending Backport

- Backport set to luminous,mimic

#3 - 02/02/2019 09:25 AM - xie xingguo

https://github.com/ceph/ceph/pull/26179

#4 - 02/04/2019 11:22 AM - Nathan Cutler

- Pull request ID set to 26179

#5 - 02/04/2019 11:22 AM - Nathan Cutler

- Copied to Backport #38162: luminous: maybe_remove_pg_upmaps incorrectly cancels valid pending upmaps added

#6 - 02/04/2019 11:22 AM - Nathan Cutler

- Copied to Backport #38163: mimic: maybe_remove_pg_upmaps incorrectly cancels valid pending upmaps added