RADOS - Bug #23646

scrub interaction with HEAD boundaries and clones is broken

04/10/2018 09:08 PM - Sage Weil

Status:	Resolved	**Start date:**	04/10/2018
Priority:	Urgent	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:	luminous	**ceph-qa-suite:**	
Regression:	No	**Component(RADOS):**	
Severity:	3 - minor	**Pull request ID:**	

Description

Scrub will work in chunks, accumulating work in cleaned_meta_map. A single object's clones may stretch across two such chunks, such that a clone is added to the cleaned_meta_map, the clone is removed, and the a later chunk finishes by getting the the head. This will result in an error like

```
2018-04-10 02:31:40.966194 7f5ca1d81700 -1 log_channel(cluster) log [ERR] : scrub 3.0 3:2525d12f::
:smithi03315943-21 ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooo:16a is an unexpected clone
```

because the cleaned_meta_map is effectively stale.

/a/yuriw-2018-04-09_19:41:27-rados-wip-yuri4-testing-2018-04-09-1710-luminous-distro-basic-smithi/2376669

Related issues:

Related to RADOS - Bug #22881: scrub interaction with HEAD boundaries and sna...	**Resolved**	**02/01/2018**
Copied to RADOS - Backport #23863: luminous: scrub interaction with HEAD boun...	**Resolved**	

History

#1 - 04/10/2018 09:09 PM - Sage Weil

- Related to Bug #22881: scrub interaction with HEAD boundaries and snapmapper repair is broken added

#2 - 04/10/2018 09:09 PM - Sage Weil

- Backport set to luminous

#3 - 04/20/2018 04:45 PM - David Zafman

We don't start trimming if scrubbing is happening, so maybe the only hole is that scrubbing doesn't check for trimming.

```
bool can_trim() {
  return
    pg->is_clean() &&
    !pg->scrubber.active &&
    !pg->snap_trimq.empty() &&
    !pg->get_osdmap()->test_flag(CEPH_OSDMAP_NOSNAPTRIM);
}
```

#4 - 04/20/2018 05:19 PM - David Zafman

I guess the intention is that scrubbing takes priority and proceeds even if trimming is in progress. Before more trim work is going to be done it stops if scrubbing is active. So the only way we could have a problem here, is that scrub might proceed with the most recent trims in flight? No more trim work will be queued once scrubber.active is set in chunky_scrub().

#5 - 04/23/2018 09:40 PM - David Zafman

- Status changed from Verified to In Progress

The osd log for primary osd.1 shows that pg 3.0 is a cache pool in a cache tiering configuration. The message "_delete_oid has or will have clones but no_whiteout=1" diagnostic could indicate a problem because we shouldn't evict a head object with any clones still there in the cache pool.

```
2018-04-10 02:31:40.048066 7f5c99570700 10 osd.1 pg_epoch: 506 pg[3.0( v 506'2158 (177'603,506'2158] local-lis
/les=253/254 n=49 ec=20/20 lis/c 253/253 les/c/f 254/258/0 253/253/20) [1,2,6] r=0 lpr=253 luod=505'2154 lua=5
06'2155 crt=506'2158 lcod 504'2153 mlcod 504'2153 active+clean+scrubbing] agent_maybe_evict evicting 3:2525d12
f:::smithi03315943-21 oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo:head(50
2'2141 osd.1.0:3764 data_digest|omap_digest s 3949843 uv 1651 dd 51da6cc7 od ffffffff alloc_hint [0 0 0])
2018-04-10 02:31:40.048071 7f5c99570700 20 osd.1 pg_epoch: 506 pg[3.0( v 506'2158 (177'603,506'2158] local-lis
/les=253/254 n=49 ec=20/20 lis/c 253/253 les/c/f 254/258/0 253/253/20) [1,2,6] r=0 lpr=253 luod=505'2154 lua=5
06'2155 crt=506'2158 lcod 504'2153 mlcod 504'2153 active+clean+scrubbing] simple_opc_create 3:2525d12f:::smith
i03315943-21 oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo:head
2018-04-10 02:31:40.048082 7f5c99570700 20 osd.1 pg_epoch: 506 pg[3.0( v 506'2158 (177'603,506'2158] local-lis
/les=253/254 n=49 ec=20/20 lis/c 253/253 les/c/f 254/258/0 253/253/20) [1,2,6] r=0 lpr=253 luod=505'2154 lua=5
06'2155 crt=506'2158 lcod 504'2153 mlcod 504'2153 active+clean+scrubbing] _delete_oid has or will have clones
but no_whiteout=1
2018-04-10 02:31:40.048087 7f5c99570700 20 osd.1 pg_epoch: 506 pg[3.0( v 506'2158 (177'603,506'2158] local-lis
/les=253/254 n=49 ec=20/20 lis/c 253/253 les/c/f 254/258/0 253/253/20) [1,2,6] r=0 lpr=253 luod=505'2154 lua=5
06'2155 crt=506'2158 lcod 504'2153 mlcod 504'2153 active+clean+scrubbing] _delete_oid 3:2525d12f:::smithi03315
943-21 oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo:head whiteout=0 no_whi
teout=1 try_no_whiteout=0
2018-04-10 02:31:40.050885 7f5c99570700 10 bluestore(/var/lib/ceph/osd/ceph-1) _remove 3.0_head #3:2525d12f:::
smithi03315943-21 oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo:head# = 0
```

#6 - 04/24/2018 12:42 AM - David Zafman

The commit below adds code to honor the no_whiteout flag even when it looks like clones exist or will exist soon. The agent_maybe_evict() used _verify_no_head_clones() to see if clones still exist.

```
  if (!snapset.clones.empty() ||
      (!ctx->snapc.snaps.empty() && ctx->snapc.snaps[0] > snapset.seq)) {
    if (no_whiteout) {
      dout(20) << __func__ << " has or will have clones but no_whiteout=1"
               << dendl;
```

```
commit de6f09f43a5213b16a9bc952e5fa092c991abb40
Author: Sage Weil <sage@redhat.com>
Date:   Fri Apr 7 13:18:50 2017 -0400

    osd/PrimaryLogPG: delete + ignore_cache is a soft hint

    We may still need to create a whiteout because clones still exist.

    Arguably delete+ignore_cache is not the right way to remove whiteouts and
    we should have a separate RADOS operation for this.  But we don't.

    Signed-off-by: Sage Weil <sage@redhat.com>
```

#7 - 04/24/2018 08:36 PM - Sage Weil

- Status changed from In Progress to Need Review

https://github.com/ceph/ceph/pull/21628

I think this will fix it?

#8 - 04/25/2018 03:58 PM - Sage Weil

- Status changed from Need Review to Pending Backport

#9 - 04/25/2018 04:00 PM - Sage Weil

master commit says:

Consider a scenario like:
- scrub [3:2525d100:::earlier:head,3:2525d12f:::foo:200]
- we see 3:2525d12f:::foo:100 and include it in scrub map

- scrub [3:2525d12f:::foo:200, 3:2525dfff:::later:head]
- some op(s) that cause scrub to be preempted
- agent_work wants to evict 3:2525d12f:::foo:100
- write_blocked_by_scrub sees scrub is preempted, returns false
- 3:2525d12f:::foo:100 is removed, :head SnapSet is updated
- scrub rescrubs [3:2525d12f:::foo:200, 3:2525dfff:::later:head]
- includes (updated) :head SnapSet
- issues error like "3:2525d12f:::foo:100 is an unexpected clone"

Fix the problem by checking if anything part of the object-to-evict and
its head touch the scrub range; if so, back off. Do not let eviction
preempt scrub; we can come back and do it later.

Fixes: http://tracker.ceph.com/issues/23646
Signed-off-by: Sage Weil <sage@redhat.com>

The same bug can happen without scrub preemption if there is a scrub chunk between the clone-to-remove and the head (i.e., object has lots of clones). Thus we should still backport this to luminous!

#10 - 04/25/2018 04:23 PM - Nathan Cutler

- Copied to Backport #23863: luminous: scrub interaction with HEAD boundaries and clones is broken added

#11 - 05/29/2018 06:49 PM - David Zafman

- Status changed from Pending Backport to Resolved