RADOS - Bug #38124

OSD down on snaptrim.

01/31/2019 12:05 PM - Darius Kasparavičius

Status:	Resolved	**Start date:**	01/31/2019
Priority:	High	**Due date:**	
Assignee:	David Zafman	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	v13.2.4
Backport:	mimic, nautilus	**ceph-qa-suite:**	
Regression:	No	**Component(RADOS):**	
Severity:	2 - major	**Pull request ID:**	27830

Description

All of ceph cluster OSD's crash when ceph runs snaptrim.

The particular error osd is throwing before crashing is:

2019-01-31 10:46:01.310 7fbb2fd45700 -1 /home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/13.2.4/rpm/el7/BUILD/ceph-13.2.4/src/osd/PrimaryLogPG.h: In function 'PrimaryLogPG::Trimming::Trimming(boost::statechart::state<PrimaryLogPG::Trimming, PrimaryLogPG::SnapTrimmer, PrimaryLogPG::WaitReservation>::my_context)' thread 7fbb2fd45700 time 2019-01-31 10:46:01.306356 /home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos7/DIST/centos7/MACHINE_SIZE/huge/release/13.2.4/rpm/el7/BUILD/ceph-13.2.4/src/osd/PrimaryLogPG.h: 1571: FAILED assert(context< SnapTrimmer >().can_trim())

```
ceph version 13.2.4 (b10be4d44915a4d78a8e06aa31919e74927b142e) mimic (stable)
 1: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0xff) [0x7fbb59f3716f]
 2: (()+0x25a337) [0x7fbb59f37337]
 3: (PrimaryLogPG::NotTrimming::react(PrimaryLogPG::KickTrim const&)+0x783) [0x559a606efbc3]
 4: (boost::statechart::simple_state&lt;PrimaryLogPG::NotTrimming, PrimaryLogPG::SnapTrimmer, boost::mpl::list&lt;mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na
, mpl_::na, mpl_::na, mpl_::na, mpl_::na, mpl_::na>, (boost::statechart::history_mode)0>::react_impl(boost::statechart::event_base const&, void const*)+0xa9) [0x559a6073b629]
 5: (boost::statechart::state_machine&lt;PrimaryLogPG::SnapTrimmer, PrimaryLogPG::NotTrimming, std::allocator&lt;void&gt;, boost::statechart::null_exception_translator>::process_queued_events()+0xb3) [0x559a60715f23]
 6: (boost::statechart::state_machine&lt;PrimaryLogPG::SnapTrimmer, PrimaryLogPG::NotTrimming, std::allocator&lt;void&gt;, boost::statechart::null_exception_translator>::process_event(boost::statechart::event_base const&)+0x87) [0x559a60716187]
 7: (_ZZN12PrimaryLogPG13WaitTrimTimerC4EN5boost10statechart5stateIS0_NS_8TrimmingENS1_3mpl4listIN4mpl_2naES8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_S8_EELNS2_12history_modeE0EE10my_contextEEN7OnTimer6finishEi()+0xb2) [0x559a607163a2]
 8: (Context::complete(int)+0x9) [0x559a60580e49]
 9: (SafeTimer::timer_thread()+0x18b) [0x7fbb59f33a8b]
 10: (SafeTimerThread::entry()+0xd) [0x7fbb59f3504d]
 11: (()+0x7dd5) [0x7fbb56ab8dd5]
 12: (clone()+0x6d) [0x7fbb55ba8ead]
 NOTE: a copy of the executable, or `objdump -rdS &lt;executable&gt;` is needed to interpret this.
```

The rest of the log is attached with the file.

Related issues:

Copied to RADOS - Backport #39698: mimic: OSD down on snaptrim.	**Resolved**

Copied to RADOS - Backport #39699: nautilus: OSD down on snaptrim.	**Resolved**

History

#1 - 02/04/2019 10:18 PM - Greg Farnum

- Project changed from Ceph to RADOS

#2 - 02/04/2019 10:23 PM - Darius Kasparavičius

- File ceph-osd.tar.gz added

Hello,

I have collected additional information Sage asked. Attached log has debug_osd=20 set.

How this happened:
1. One of the nodes had all it's osd's set to out. To clean them up for replacement.
2. Noticed that a lot of snaptrim was running.
3. Set nosnaptrim flag on the cluster.
4. Once mon_osd_snap_trim_queue_warn_on appeared. Removed nosnaptrim flag.
5. All osds on the cluster crashed and started flapping. Set nosnaptrim flag back on.

#3 - 02/06/2019 10:13 PM - Neha Ojha

- Priority changed from Normal to High

#4 - 02/06/2019 10:42 PM - Greg Farnum

I was theorizing in a bug scrub that maybe the PG was running behind on OSDMaps and so missing the nosnaptrim flag update, but that isn't the case — the OSD doesn't look at it directly at all, just the PG when it activates a map.

However, since the crash came from the WaitTrimTimer state's timer triggering a transition into NotTrimming and posting a KickTrim event, I think it's safe to say there's some race or missed timer cleanup that causes this when the flag changes state. Since the timer is cleaned up when you exit the WaitTrimTimer state, that also seems a bit odd, but maybe...oh, in fact, I don't see anything that directly kills it. Maybe we do a wide reset() somewhere? Kinda looks like we only do that in PrimaryLogPG::on_change() and that is only triggered in interval changes, though.

(Also interesting: the WaitRepops state goes into WaitTrimTimer if !can_trim() when the replies come back.)

#5 - 03/08/2019 08:19 AM - Darius Kasparavičius

Hello,

any updates regarding this bug? I would love a patch to resolve this issue ASAP. One of my monitors just died and I can't add new one. As it's throwing slow io errors while trying to synchronise.

#6 - 03/18/2019 02:22 PM - Erikas Kučinskis

Hello any updates about this?

#7 - 04/02/2019 01:02 PM - Erikas Kučinskis

Hello it's been two months now is there any update about this bug?

#8 - 04/26/2019 11:19 PM - David Zafman

- Status changed from New to In Progress

- Assignee set to David Zafman

I am able to reproduce this, so I'll work on a fix.

#9 - 04/27/2019 11:26 PM - David Zafman

The following script sometimes hits the race and crashes an OSD. I've removed the assert and the script has been running in a loop without seeing any other core dumps.

```
#! /bin/bash -x

../src/stop.sh
MGR=1 MON=1 MDS=0 OSD=5 ../src/vstart.sh -l -d -n -o osd_snap_trim_sleep=5.0 2> /dev/null
sleep 5
bin/ceph osd pool create test 1 1 2> /dev/null
sleep 5

sleep 2
bin/ceph pg dump pgs 2> /dev/null

for s in $(seq 1 20)
do
  dd if=/dev/urandom bs=1m count=1 of=data
  for i in $(seq 1 100)
  do
    bin/rados -p test put obj$i data 2> /dev/null
  done
  bin/rados -p test mksnap snap${s} 2> /dev/null
done

while(true); do bin/ceph osd set nosnaptrim; sleep 1; bin/ceph osd unset nosnaptrim; done &

for s in $(seq 1 20)
do
  bin/rados -p test rmsnap snap$s
  sleep 3
done

sleep 60
bin/ceph status
bin/ceph osd dump

kill %%
wait
bin/ceph status
bin/ceph osd dump
```

#10 - 04/28/2019 12:25 AM - David Zafman

- Pull request ID set to 27830

#11 - 04/28/2019 12:29 AM - David Zafman

- Status changed from In Progress to Need Review

#12 - 05/07/2019 10:15 AM - Erikas Kučinskis

Hi is there any ETA when the bug will be live?

#13 - 05/07/2019 10:16 AM - Erikas Kučinskis

Erikas Kučinskis wrote:

> Hi is there any ETA when the bug fix will be live?

#14 - 05/08/2019 09:33 PM - Greg Farnum

- Status changed from Need Review to Pending Backport

- Backport set to mimic, nautilus

No ETA; it'll have to wend its way through the backports process. I don't think any releases are imminent so it should be the next point release though.

#15 - 05/09/2019 07:10 AM - Erikas Kučinskis

Greg Farnum wrote:

> No ETA; it'll have to wend its way through the backports process. I don't think any releases are imminent so it should be the next point release though.

Thank you for the information.

#16 - 05/10/2019 11:00 AM - Nathan Cutler

- Copied to Backport #39698: mimic: OSD down on snaptrim. added

#17 - 05/10/2019 11:00 AM - Nathan Cutler

- Copied to Backport #39699: nautilus: OSD down on snaptrim. added

#18 - 07/12/2019 12:33 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved

Files

ceph-log.zip	282 KB	01/31/2019	Darius Kasparavičius

ceph-osd.tar.gz	754 KB	02/04/2019	Darius Kasparavičius