bluestore - Bug #41037

Containerized cluster failure due to osd_memory_target not being set to ratio of cgroup_limit per osd_memory_target_cgroup_limit_ratio

07/31/2019 07:21 PM - Dustin Black

Status:	Resolved	**% Done:**	0%
Priority:	Urgent		
Assignee:			
Category:			
Target version:	v14.2.3		
Source:		**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	nautilus	**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	2 - major	**Crash signature (v2):**	
Reviewed:			

Description

Under heavy I/O workload (generated to multiple postgres databaseses, backed by Ceph RBD, via the pgbench utility), we have experienced a condition where the memory limit for the OSD pods is reached, OOM kills happen, and the cluster then becomes unresponsive. This is a critical failure caused apparently by the osd_memory_target parameter not being sized with some headroom below the cgroup_limit. Mark Nelson has repeatedly said that 10-20% headroom was needed to ensure that the OSD could trim its memory before the OOM killer was triggered.

Note that because Bluestore is computing the osd_memory_target for us based on the CGroup limit, there is no way to override this, so this is a high priority problem!

Deploying containerized Ceph via Rook.io, we apply a memory limit to the OSDs via the cluster.yaml, but *not* a memory request.

```
  resources:
    osd:
      requests:
        cpu: "2"
      limits:
        cpu: "2"
        memory: "6Gi"
```

Per fc3bdad [1], our expectation is that, without a memory request, the value of osd_memory_target should default to cgroup_limit * osd_memory_target_cgroup_limit_ratio (0.8 default). However, the deployed cluster shows ...

```
sh-4.2# ceph daemon /var/lib/rook/osd0/ceph-osd.0.asok config show | grep osd_memory
    "osd_memory_base": "805306368",
    "osd_memory_cache_min": "134217728",
    "osd_memory_cache_resize_interval": "1.000000",
    "osd_memory_expected_fragmentation": "0.150000",
    "osd_memory_target": "6442450944",  <-- Target = limit instead of = (limit*0.8)
    "osd_memory_target_cgroup_limit_ratio": "0.800000",  <-- Ratio looks correct
```

```
  cluster:
    id:     ffa396e4-7874-472d-95fa-692d754f5e6e
    health: HEALTH_ERR
            1 MDSs report slow metadata IOs
            2 osds down
```

```
            15/76461 objects unfound (0.020%)
            Reduced data availability: 309 pgs inactive, 309 pgs peering, 103 pgs stale
            Possible data damage: 10 pgs recovery_unfound
            Degraded data redundancy: 73406/229339 objects degraded (32.008%), 669 pgs degraded, 6
44 pgs undersized

  services:
    mon: 3 daemons, quorum a,b,c (age 2w)
    mgr: a(active, since 2d)
    mds: myfs:1 {0=myfs-b=up:active} 1 up:standby-replay
    osd: 12 osds: 6 up (since 24m), 8 in (since 23m); 104 remapped pgs

  data:
    pools:   3 pools, 1224 pgs
    objects: 76.46k objects, 296 GiB
    usage:   638 GiB used, 11 TiB / 12 TiB avail
    pgs:     25.245% pgs not active
             73406/229339 objects degraded (32.008%)
             15/76461 objects unfound (0.020%)
             548 active+undersized+degraded
             253 peering
             145 active+clean
             100 stale+active+clean
             56  remapped+peering
             40  active+recovery_wait+undersized+degraded
             36  active+undersized+degraded+remapped+backfill_wait
             23  active+recovery_wait+degraded
             9   active+recovery_wait+undersized+degraded+remapped
             6   active+recovery_unfound+undersized+degraded+remapped
             4   active+recovery_unfound+undersized+degraded
             2   stale+active+recovery_wait+degraded
             1   stale+active+undersized+degraded
             1   active+recovery_wait
```

[1] https://github.com/ceph/ceph/commit/fc3bdad87597066a813a3734b2a79e803340be36#diff-a9faffcf40600fd57aea5451cef5abe9

Related issues:

Copied to bluestore - Backport #41273: nautilus: Containerized cluster failur...	**Resolved**

History

#1 - 07/31/2019 09:07 PM - Neha Ojha

Which version of Ceph are you running?

#2 - 07/31/2019 09:25 PM - Neha Ojha

Can you enable debug_osd=10 and see what this line (https://github.com/ceph/ceph/commit/fc3bdad87597066a813a3734b2a79e803340be36#diff-a9faffcf40600fd57aea5451cef5abe9R4214) reports in the log?

#3 - 08/01/2019 05:38 PM - Ben England

- Target version set to v14.2.3

version you asked for:

ceph-base-14.2.2-0.el7.x86_64

from the Ceph container image ceph/ceph:v14.2.2-20190722

#4 - 08/01/2019 05:44 PM - Ben England

from Joe T on his system:

1. ceph version
   ceph version 14.2.2-218-g734b519 (734b5199dc45d3d36c8d8d066d6249cc304d0e0e) nautilus (stable)

#5 - 08/01/2019 07:17 PM - Neha Ojha

Neha Ojha wrote:

> Can you enable debug_osd=10 and see what this line (https://github.com/ceph/ceph/commit/fc3bdad87597066a813a3734b2a79e803340be36#diff-a9faffcf40600fd57aea5451cef5abe9R4214) reports in the log?

my bad, it should be debug_bluestore=10

#6 - 08/01/2019 08:36 PM - Josh Durgin

- Priority changed from Normal to Urgent

Joe Talerico reproduced this and found the POD_LIMIT was getting set, but not the system-wide limit, so the current OSD code reading only the system limit would have no effect.

These info available in the osd container was:

```
cat /proc/80946/cgroup
12:pids:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.sl
ice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
11:perf_event:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616
b98.slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
10:devices:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98
.slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
9:hugetlb:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.
slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
8:freezer:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.
slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
7:blkio:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.sl
ice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
6:cpu,cpuacct:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616
b98.slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
5:memory:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.s
lice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
4:rdma:/
3:net_cls,net_prio:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_9803
9b616b98.slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
2:cpuset:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b616b98.s
lice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope
1:name=systemd:/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b1_98039b61
6b98.slice/crio-7d8c95780ecfe701043003a914b4ab2cb410c0139478e46db8f07008ba8e733e.scope

cat /sys/fs/cgroup/memory/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod688470ce_b491_11e9_b8b
1_98039b616b98.slice/memory.limit_in_bytes
6442450944

cat /sys/fs/cgroup/memory/memory.limit_in_bytes
9223372036854771712
```

#7 - 08/05/2019 02:09 PM - Ben England

My guess would be if CGroup limit is X, then 0.95 X - 1/2 GB should be fine for osd_memory_target, that would give the OSDs time to detect that they are over their limit and purge the cache to bring it back down, while wasting as little memory resource as possible.

The worst case here is really intense I/O to NVM devices, where the cache size can increase rapidly before the OSD can detect that there is a problem. In the past, HDDs limited the IOPS rate.

#8 - 08/05/2019 04:05 PM - Mark Nelson

@ben that's probably a semi-reasonable assumption in a lot of cases, though I've noticed that the kernel doesn't always reclaim unmapped memory right away which can make this really tricky. CentOS and RHEL seem to do better than Ubuntu does though and I have no idea how containers interact with it.

Typically I just say to give them an extra 20% but I could be convinced something like 5%+XMB could also work.

#9 - 08/06/2019 04:16 PM - Sage Weil

- Status changed from New to Fix Under Review

- Backport set to nautilus

https://github.com/ceph/ceph/pull/29511

#10 - 08/08/2019 08:52 PM - Sage Weil

- Status changed from Fix Under Review to Pending Backport

nautilus backport: https://github.com/ceph/ceph/pull/29562

#11 - 08/15/2019 09:08 AM - Nathan Cutler

- Copied to Backport #41273: nautilus: Containerized cluster failure due to osd_memory_target not being set to ratio of cgroup_limit per osd_memory_target_cgroup_limit_ratio added

#12 - 08/27/2019 02:54 PM - Josh Durgin

- Status changed from Pending Backport to Resolved