# II: SYSTEMS

# **Troubleshooting Ceph**

Tips & Tricks

Cory Snyder



## About me/11:11 Systems

- Principal Software Engineer on the SRE team at 11:11 Systems
- 11:11 Systems is a managed infrastructure solutions provider with a global footprint that is well known for backup and DRaaS, but also has many other offerings in the cloud, connectivity, and security space
- Joined by colleagues David Orman and Welby McRoberts
- David is on the Ceph leadership team

## **Overview**

- Discuss major issues that we've encountered over the past 4 years and associated troubleshooting takeaways
- How to get help from the community and developers
- What we can do better



## **Major Issues**

- Manager deadlock bug
- rocksdb tombstone iteration performance bugs
- RGW versioned bucket index issues

## Manager deadlock bug

#### Symptoms

- Manager unresponsive python modules not executing
- Manager logs go quiet
- Happened randomly, anywhere from a few minutes to a few hours after a manager became active



## Manager deadlock bug

#### Diagnosis

- Installed relevant debuginfo, source packages, development tool packages, python gdb support in container built on top of release container
- Attached to the hung mgr process with gdb and examined the backtrace of each active thread
- Found that the thread with the GIL was waiting on another mutex.
- Used gdb to search through other threads to find the one that held this mutex.
- Found that the thread which held the mutex was waiting on the GIL
- Analyzed CPython code and backtraces to discover that Python may internally give up the GIL while performing GC

II:II SYSTEMS

• The solution to this deadlock scenario is to always release the GIL before attempting to acquire a mutex.



## Manager deadlock bug

#### Takeaways

- gcore + gdb
- GDB python support:
  - https://devguide.python.org/development-tools/gdb/
- Debug container:
  - run with --privileged, --pid=host
  - When using ebpf, also sometimes want to include:
    - -v /lib/modules:/lib/modules:ro
    - -v /sys/kernel:/sys/kernel:ro
    - -v /usr/src:/usr/src:ro

RUN dnf group install -y "Development Tools"
RUN dnf install -y \
wget \
bpftrace \
perf \
glibc-langpack-en 🔪
glibc-locale-source
RUN dnf install -y \
ceph-base-debuginfo 🔪
ceph-common-debuginfo \
ceph-osd-debuginfo \
ceph-mgr-debuginfo
ceph-mon-debuginfo \
ceph-radosgw-debuginfo
# python3-rados-debuginfo \ # python3-rbd-debuginfo \
# python3-rgw-debuginfo
# librados2-debuginfo \
# librbd1-debuginfo \
# librgw2-debuginfo \
# ceph-fuse-debuginfo \
# ceph-immutable-object-cache-debuginfo \
# ceph-mds-debuginfo \
# cephfs-mirror-debuginfo \
# libcephfs2-debuginfo \
# libcephsglite-debuginfo \
# libradosstriper1-debuginfo \
# python3-cephfs-debuginfo \
# rbd-fuse-debuginfo \
# rbd-mirror-debuginfo \
# rbd-nbd-debuginfo
WORKDIR root
RUN wget https://github.com/cfsnyder/wachy/releases/download/0.1.0-alpha.7 search fix/wach

## rocksdb tombstone iteration issues

#### Symptoms

- Slow ops
- OSD Suicide timeouts
- Extreme client latencies, poor throughput

#### Two separate incidents:

- RGW bucket post-reshard cleanup
- Large PG movement

241 slow ops, oldest one blocked for 301 sec, daemons [osd.103,osd.106,osd.107,osd.108,osd.78,osd.84] have slow ops.

client: 410 KiB/s rd, 110 KiB/s wr, 165 op/s rd, 68 op/s wr

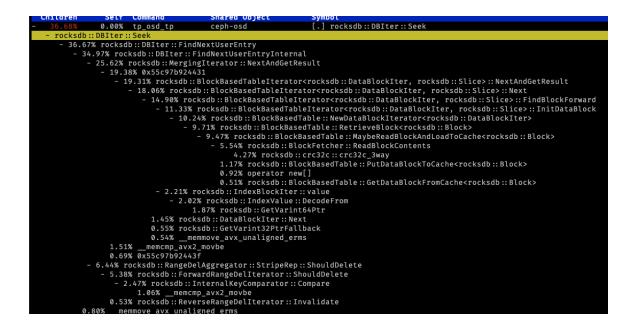
debug 2022-03-10T09:55:23.939+0000 7f905fc87700 0 log\_channel(cluster) log [WRN] : slow request osd\_op(client.768029.0:219375 23.7b 23:de197397:::.dir.6d3ebb2d-abcd-44c1-8d25-da659648a7a1.687487.1.154:head [call rgw.bucket\_list in=255b] snapc 0=[] ondisk+read+known\_if\_redirected e28677) initiated 2022-03-10T09:17:00.528266+0000 currently delayed

debug 2022-03-10T10:25:52.211+0000 7f684ab37700 0 bluestore(/var/lib/ceph/osd/ceph-103) log\_latency\_fn slow operation observed for upper\_bound, latency = 5.113506794s, after = omap\_iterator(cid = 23.7b\_head, o id = #23:de197397:::.dir.6d3ebb2d-abcd-44c1-8d25-da659648a7a1.687487.1.154:head#)

## rocksdb tombstone iteration issues

#### Diagnosis

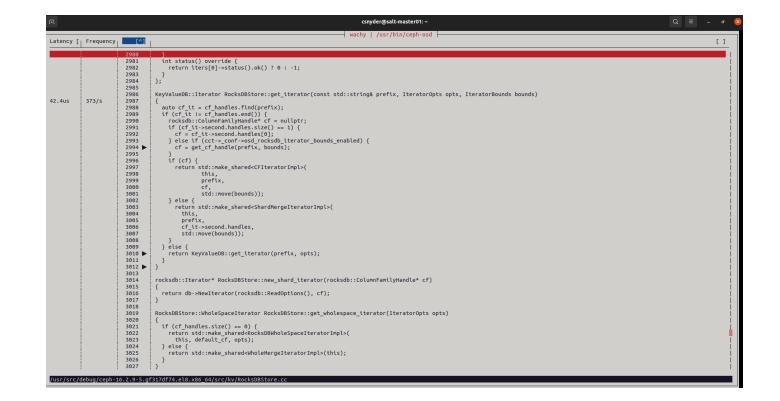
- Centralized logging with Loki and Grafana to help identify problematic OSDs
- OSD logs indicating slow ops, slow function execution in bluestore
- Debug container
- perf command
- wachy ebpf tool
- Export rocksdb database with ceph-bluestoretool
- Source code analysis
- Confirmed hypothesis by validating manual compaction relieved issues



## rocksdb tombstone iteration issues

#### Takeaways

- perf
- wachy
- Centralized logging and metrics
- ceph-ci for container builds
- ceph-kvstore-tool
  - manually compact rocksdb
- ceph-bluestore-tool
  - bluefs-export
- ceph-objectstore-tool
  - directly inspect object store data



## **RGW versioned bucket index bugs**

#### Symptoms

- S3 bucket listings not completing or extremely slow
- RGW lifecycle processing failing
- Clients seeing 200 response with empty body for keys that should give 404

## **RGW versioned bucket index bugs**

#### Diagnosis

- radosgw-admin bi list
- Inspecting raw rados object xattrs, omap entries, and data with rados command
- Verbose debug logs
- Initially wrote custom python scripts to inspect bucket index entries and identify those that weren't quite right
- Hypothesis testing and reproducers with vstart cluster
- Modified radosgw-admin tool with new commands to find and fix problems

## **RGW versioned bucket index bugs**

#### Takeaways

- It pays to be familiar with how RGW stores data
  - omap and omap header
  - xattrs
  - rados object content
- Using a vstart cluster to play with code, test hypotheses, and develop reproducers can be very effective

## How to get help

- Email lists
- Slack / IRC
- Weekly ceph meetups:
  - https://ceph.io/en/community/meetups/



## Gathering good debugging info

- Daemon logs, enable higher debug levels >= 20
- perf / wachy / ebpf stats
- System state output from relevant ceph commands
- Core dumps
- Test cases as reproducers

## What can we do better?

- Official debug containers?
- Tracing in production?
- More fine-grained control over logging?
- Tool to help with obfuscating logs?
- Tools to export diagnostic data that can be used to rapidly identify problems?
- Better real time insights into where bottlenecks might exist, perhaps integrated into ceph health module?
- Faster container builds / local container builds / faster container deployments?

# 

1111SYSTEMS.COM

II:II SYSTEMS

RETHINK CONNECTED