

||:|| 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
- 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

```
1 FROM quay.io/ceph/ceph:v16.2.14
2
3 RUN dnf group install -y "Development Tools"
4 RUN dnf install -y \
5     wget \
6     bpftrace \
7     perf \
8     glibc-langpack-en \
9     glibc-locale-source
10 RUN dnf install -y \
11     ceph-base-debuginfo \
12     ceph-common-debuginfo \
13     ceph-osd-debuginfo \
14     ceph-mgr-debuginfo \
15     ceph-mon-debuginfo \
16     ceph-radosgw-debuginfo
17 # python3-rados-debuginfo \
18 # python3-rbd-debuginfo \
19 # python3-rgw-debuginfo
20 # librados2-debuginfo \
21 # librbid1-debuginfo \
22 # librgw2-debuginfo \
23 # ceph-fuse-debuginfo \
24 # ceph-inmutable-object-cache-debuginfo \
25 # ceph-nds-debuginfo \
26 # cephfs-mirror-debuginfo \
27 # libcephfs2-debuginfo \
28 # libcephsqlite-debuginfo \
29 # libradosstriper1-debuginfo \
30 # python3-cephfs-debuginfo \
31 # rbd-fuse-debuginfo \
32 # rbd-mirror-debuginfo \
33 # rbd-nbd-debuginfo
34
35 WORKDIR root
36 RUN wget https://github.com/cfsnyder/wachy/releases/download/0.1.0-alpha.7_search_fix/wachy
37 RUN chmod +x wachy
38 RUN export LC_ALL="en_US.UTF-8"
39 RUN localedef -v -c -i en_US -f UTF-8 en_US.UTF-8 || :
```

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] snapc0=[]  
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-bluestore-tool
- Source code analysis
- Confirmed hypothesis by validating manual compaction relieved issues

```
Children      Self  Command      Shared Object      Symbol
- 36.68%      0.00% tp_osd_tp      ceph-osd           [.] rocksdb::DBIter::Seek
- rocksdb::DBIter::Seek
- 36.67% rocksdb::DBIter::FindNextUserEntry
- 34.97% rocksdb::DBIter::FindNextUserEntryInternal
- 25.62% rocksdb::MergingIterator::NextAndGetResult
- 19.38% 0x55c97b924431
- 19.31% rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::NextAndGetResult
- 18.06% rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::Next
- 14.90% rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::FindBlockForward
- 11.33% rocksdb::BlockBasedTableIterator<rocksdb::DataBlockIter, rocksdb::Slice>::InitDataBlock
- 10.24% rocksdb::BlockBasedTable::NewDataBlockIterator<rocksdb::DataBlockIter>
- 9.71% rocksdb::BlockBasedTable::RetrieveBlock<rocksdb::Block>
- 9.47% rocksdb::BlockBasedTable::MaybeReadBlockAndLoadToCache<rocksdb::Block>
- 5.54% rocksdb::BlockFetcher::ReadBlockContents
  4.27% rocksdb::crc32c::crc32c_3way
  1.17% rocksdb::BlockBasedTable::PutDataBlockToCache<rocksdb::Block>
  0.92% operator new[]
  0.51% rocksdb::BlockBasedTable::GetDataBlockFromCache<rocksdb::Block>
- 2.21% rocksdb::IndexBlockIter::value
- 2.02% rocksdb::IndexValue::DecodeFrom
  1.87% rocksdb::GetVarint64Ptr
  1.45% rocksdb::DataBlockIter::Next
  0.55% rocksdb::GetVarint32PtrFallback
  0.54% __memmove_avx_unaligned_erms
  1.51% __memcmp_avx2_movbe
  0.69% 0x55c97b92443f
- 6.44% rocksdb::RangeDelAggregator::StripeRep::ShouldDelete
- 5.38% rocksdb::ForwardRangeDelIterator::ShouldDelete
- 2.47% rocksdb::InternalKeyComparator::Compare
  1.06% __memcmp_avx2_movbe
  0.53% rocksdb::ReverseRangeDelIterator::Invalidate
  0.80% memmove_avx_unaligned_erms
```

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

```
csnyder@salt-master01: ~
wachy | /usr/bin/ceph-osc
[ ]
Latency [ Frequency [ ^
42.4us 373/s
2980 }
2981 int status() override {
2982     return iters[0]->status().ok() ? 0 : -1;
2983 }
2984 };
2985
2986 KeyValueDB::Iterator RocksDBStore::get_iterator(const std::string& prefix, IteratorOpts opts, IteratorBounds bounds)
2987 {
2988     auto cf_it = cf_handles.find(prefix);
2989     if (cf_it != cf_handles.end()) {
2990         rocksdb::ColumnFamilyHandle* cf = nullptr;
2991         if (cf_it->second.handles.size() == 1) {
2992             cf = cf_it->second.handles[0];
2993         } else if (cct->conf->osd_rocksdb_iterator_bounds_enabled) {
2994             cf = get_cf_handle(prefix, bounds);
2995         }
2996         if (cf) {
2997             return std::make_shared<CFIteratorImpl>(
2998                 this,
2999                 prefix,
3000                 cf,
3001                 std::move(bounds));
3002         } else {
3003             return std::make_shared<ShardMergeIteratorImpl>(
3004                 this,
3005                 prefix,
3006                 cf_it->second.handles,
3007                 std::move(bounds));
3008         }
3009     } else {
3010         return KeyValueDB::get_iterator(prefix, opts);
3011     }
3012 }
3013
3014 rocksdb::Iterator* RocksDBStore::new_shard_iterator(rocksdb::ColumnFamilyHandle* cf)
3015 {
3016     return db->NewIterator(rocksdb::ReadOptions(), cf);
3017 }
3018
3019 RocksDBStore::WholeSpaceIterator RocksDBStore::get_wholespace_iterator(IteratorOpts opts)
3020 {
3021     if (cf_handles.size() == 0) {
3022         return std::make_shared<RocksDBWholeSpaceIteratorImpl>(
3023             this, default_cf, opts);
3024     } else {
3025         return std::make_shared<WholeMergeIteratorImpl>(this);
3026     }
3027 }
/usr/src/debug/ceph-16.2.9-5.gf317df74.el8.x86_64/src/kv/RocksDBStore.cc
```

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?

THANK YOU

1111SYSTEMS.COM