Configuring Ceph

(An incomplete guide, v0.1-2016-12-05)

Dan van der Ster
CERN IT Storage Group

Overview

- Ceph has many config options: ~860 in hammer, ~1100 in jewel
- There are **no silver bullets**:
  - The defaults are intended to handle the general case: be pretty good for most clusters.
    - Though in my experience some defaults are *incorrect*
  - So many options exist because there is not always a good default for every cluster.

- Outline:
  - First, we'll see how to get and set options.
  - Next, what's in our `ceph.conf`
    - `[global]`
    - `[mon]`
    - `[osd]`
    - `[mds]`
    - (`[radosgw]` not covered in this talk)

- Please double check docs/ML before applying any of these configs to your cluster.
- BTW, this talk is not at all about CRUSH (incl. CRUSH trees, rules, pool options, …)

Which options are available?

- Online documentation is pretty OK:
  - http://docs.ceph.com/docs/master/
  - OSD config: http://docs.ceph.com/docs/master/rados/configuration/osd-config-ref/
  - **Pay close attention to which release you're looking at: master/jewel/hammer**

- Occasionally the docs are incomplete; there are two definitive ways to see which options exist:
  - Read the source code: https://github.com/ceph/ceph/blob/jewel/src/common/config_opts.h
  - Ask the daemons directly…

Querying Ceph Daemons Directly

- ceph daemon <type.id> config <diff|get|set|show>

```
"config diff": "dump diff of current config and default config"
"config get": "config get <field>: get the config value"
"config set": "config set <field> <val> [<val> ...]: set a config variable"
"config show": "dump current config settings”
```

- Example: Which options exist for osd scrubbing?

```
# ceph daemon osd.0 config show | grep scrub
"osd_scrub_thread_timeout": "60",
"osd_scrub_thread_suicide_timeout": "300",
"osd_scrub_finalize_thread_timeout": "600",
"osd_scrub_invalid_stats": "true",
"osd_max_scrubs": "1",
…
```

How does my config differ from the default?

- It is sometimes useful to understand how a daemon differs from the default configuration:

```
# ceph daemon osd.0 config diff
{    "diff": {        "current": { … },        "defaults": { … }    }}
```

- Example: How does my osd thread config differ from the default?

```
# ceph daemon osd.0 config diff | grep thread
"filestore_op_threads": "8",
"internal_safe_to_start_threads": "true",
"osd_disk_thread_ioprio_class": "idle",
"osd_disk_thread_ioprio_priority": "0",
"osd_op_thread_suicide_timeout": "300",
"osd_op_threads": "12",
```

How to set/change options?

- Options should be set in `/etc/ceph/ceph.conf`.
- How to test new options without restarting daemons?
- Two methods for changing a single daemon:

```
ceph daemon osd.0 config set osd_scrub_sleep 0.1
ceph tell osd.0 injectargs -- --osd_scrub_sleep=0.1
```

- How to set many options cluster-wide:

```
ulimit -n 10000; ceph tell osd.* injectargs -- --osd_scrub_sleep=0.1 --osd_max_scrubs=1
```

- Not all options can be set at run-time. (Some only take effect after a reboot).
- Options set at run-time are not automatically persisted to `ceph.conf`.

- General advice: **keep `ceph -w` open in another window when changing options**.
  - If your change causes problems, you'll quickly notice, then you can quickly revert.

`ceph.conf`: how does it work?

- Different sections for the each daemon. `[mon]` `[osd]` `[mds]` …

- Sections evaluated in priority order:
  - `[type.id]` > `[type]` > `[global]`
  - Example `osd.0` reads options from `[osd.0]`, then `[osd]`, then `[global]`.
  - IOW:
    - `[global]` applies to all daemons/clients
    - `[mon]` applies to all mons (but not osds/mds's)
    - `[mon.p01001532021656]` applies only to `mon.p01001532021656`.

CERN's `ceph.conf`

Again, I don't recommend to copy these blindly.

These are just my best hints for shrinking the full parameter space down to the most meaningful options.

[global]

- Essential:
```
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
fsid = xxx
```
- Debugging, disabled for the common daemons:
```
debug filestore = 0
debug mon = 0
debug osd = 0
```
- A couple mon options:
```
mon compact on start = true # useful to keep the mon leveldb small
mon osd down out interval = 900  # allows for reboots without backfilling
```
- One ms option:
```
ms tcp read timeout = 60 # much shorter than the default 900s – fixes a bug in old el6 kernel
```
- Smaller osdmap footprint (see later slide)
```
osd map message max = 100 # part of the recipe for smaller osdmap memory footprint
```

[mon]

- Nothing too insightful here:

```
mon osd down out subtree limit = host

mon pg warn max object skew = 0
mon pg warn min per osd = 0

osd pool default flag hashpspool = true

osd pool default size = 3
osd pool default min size = 2

osd pool default pg num = 64
osd pool default pgp num = 64
```

[osd] basics

- Some basic options to get an OSD started

```
journal size = 20480  # disk throughput / filestore max sync interval

osd mkfs type = xfs
osd mount options xfs = rw, noatime, inode64, logbufs=8, logbsize=256k

crush location = room=0513-R-0050 rack=RA05 host=p06253939q56782

osd crush update on start = false # know when to set this to true or false
```

[osd] tuning, threads

- General options to tune the OSD concurrency

```
filestore max sync interval = 60 # fsync files every 60s
filestore fd cache size = 2048
filestore op threads = 8  # more threads where needed
filestore queue max ops = 100 # allow more queued ops

max open files = 65536 # change the ulimit to open more files

osd disk threads = 1
osd op threads = 12  # more threads where needed
osd op thread suicide timeout = 300 # increased to survive router reboots
osd snap trim sleep = 0.1   # throttle some long lived OSD ops
```

[osd] scrubbing

- We have an ongoing problem to minimize the impact of background scrubbing on client latency.

```
osd max scrubs = 1
osd scrub max interval = 4838400
osd scrub min interval = 2419200
osd deep scrub interval = 2419200
osd scrub interval randomize ratio = 1.0

osd disk thread ioprio class = idle
osd disk thread ioprio priority = 0

osd scrub chunk max = 1
osd scrub chunk min = 1
osd deep scrub stride = 1048576
osd scrub load threshold = 5.0
osd scrub sleep = 0.1
```

[osd] backfilling and recovery

- Likewise we always try to throttle recovery/backfilling as much as possible:

```
osd max backfills = 1
osd recovery max active = 1
osd recovery max single start = 1
osd recovery op priority = 1
osd recovery threads = 1
osd backfill scan max = 16
osd backfill scan min = 4
```

[mds] (WIP)

- Not much tuning here:

```
debug mds = 2  # print some useful stats – not very verbose.
mds standby replay = true  # keep the standby hot
mds cache size = 3209947  # $::memorysize_mb * 50
```

Cookbook: Minimizing OSD map footprint

- The osdmap cache can consume a lot of RAM for every OSD. Jewel lowers the size of that cache from 500 to 200 entries. (below are the jewel values – safe to apply these to hammer cluster)
- Consider dividing all values below by 10 in case of a memory crisis [1]

```
[global]
    osd map message max = 100

[osd]
    osd map cache size = 200
    osd map max advance = 150
    osd map share max epochs = 100
    osd pg epoch persisted max stale = 150
```

[1] https://cds.cern.ch/record/2015206/files/CephScaleTestMarch2015.pdf

sysctl.conf

```
kernel.pid_max = 4194303
vm.zone_reclaim_mode = 0
vm.swappiness = 10
vm.min_free_kbytes = 513690 # assuming 64GB of RAM total
vm.dirty_ratio = 40
vm.vfs_cache_pressure = 100
net.netfilter.nf_conntrack_max = 10000000
net.nf_conntrack_max = 10000000
```