Ceph in the GRNET cloud stack

Nikos Kormpakis

September 26, 2017

GRNET - Greek Research and Technology Network

About GRNET

GRNET’s Infrastructure

- 5 datacenters (3 in Athens, 1 in Louros, 1 in Crete)
- ~1000 VMCs (Virtual Machine Containers)
- ~180 SCs (Storage Containers, only for Ceph)
- 4 NetApp boxes (3x FAS8040, 1x IBM N5300)
- EMC boxes, IBM TS4500 tape library etc etc

GRNET Cloud Platform

- ViMa: VPS for GRNET Peers (not really ”cloud”)
  - ganetimgr
  - NFS and DRBD
- ~okeanos: Elastic VMs for students, researchers
  - Synnefo
  - DRBD, RBD, Archipelago (Ceph)

30 Ganeti clusters, QEMU/KVM, Debian Jessie and a lot of more…

Ceph @ GRNET

Ceph Infrastructure

4 Ceph clusters

- 2 production clusters
- 2 testing clusters

Some facts:

- Each ~okeanos installation has its own cluster
- Variety of hardware
- Spine-leaf network topology
- Each cluster lives only in one DC
- Mix of Ceph versions and setups
- No NAT, everything dualstack

Where do we use Ceph?

Two large use cases

- Block Storage for VMs
  ~okeanos (and maybe later for ViMa), using Archipelago and RBD.
- Pithos+
  dropbox-like object storage, part of ~okeanos, using Archipelago.

Ceph Clusters

rd0 Cluster

rd0, old cluster (~okeanos)

- ~**300**TB raw storage, **164** OSDs, **3** MONs, **6** OSDs/node
- MONs colocated with OSDs
- **HP ProLiant DL380 G7** with 2x **HP DS2600** enclosures/node
- SAS disks for journals and OSDs, all on **RAID1**...
- **Hammer** (v0.94.9), **3.16** kernel, with default crushmap
- No more space allocated, will be deprecated
- Used only for Archipelago (more on it later)
- Started 4 years ago as an experiment for Archipelago

rd0 Cluster

What's wrong with this cluster?

- All disks are on **RAID1** -> low performance, 50% space loss
- **Broadcom NetXtreme II BCM5709** NICs really suck
  A lot of flapping has caused multiple outages (1 major one)
- replica count = 2, min_size=1...
- **No deep-scrub** due to performance issues
- Non-optimal tunables set
- **No warranty**
- ext4 OSDs

rd0 Cluster

About the future of rd0

- Old, but **functional** hardware
- Suitable for users with little or no I/O performance requirements (i.e. short-living VMs for student classes)
- Install refurbished **X540** NICs and get rid of the old ones
- Upgrade to **Jewel** yesterday
- Recreate all OSDs with **ceph-deploy** (XFS)
- Add more nodes to cluster
- Increase replica count to **3**, set optimal tunables
- Maybe... use **single disk OSDs** instead of RAID1

rd1 Cluster

rd1, new Ceph production cluster (~okeanos-knossos)

- ~**550**TB raw storage, **144** OSDs, **3** MONs, **12** OSDs/node
- **Lenovo ThinkServer RD550** with SA120 arrays (JBOD enabled)
- **SSD** for MON stores, **12x4TB** SAS for OSD stores, **6x200GB** for OSD journals (SSDs)
- **2x Journals/OSD**, 20GB per Journal
- 2x Intel Xeon E5-2640V3 2.60GHz, 128GB RAM
- 2x10G LACP
- **Jewel** (v10.2.9), **4.9** kernel, default crushmap, optimal tunables

rd1 Cluster

Configuration, tuning etc

- **RBD** and **Archipelago**
- **librbd** for VMs (more on krbd vs librbd)
- MONs colocated with OSDs. Each MON on different rack.
- sysctl && controller settings (tcp stuff, pid/thread limits, perf governor etc)
- Configs mostly from ML suggestions (+testing of course)
- deep-scrub enabled with low priority settings
- **Cephx** auth enabled
- Separate client and replication networks

rd1 Cluster

Pain points

- Large number of threads with librbd images
  Considering using async messenger
- Trouble with systemd/udev (had to upgrade to systemd v230)
- Issues with DC network hardware cause flaps, packet loss, etc
- (Almost) no monitoring for (lib)rbd on clients, working on it
- There’s still a lot to learn about RBD

OK, what is Archipelago?

Archipelago

Archipelago is a storage layer that provides a unified way to handle volumes/files, independently of the storage backend.

- C
- Uses blktap
- Two backends supported: NFS and RADOS (librados)
- Nothing needed on Ceph's side, just two pools: `blocks` and `maps`
    - `blocks` pool used for actual disk blocks
    - objects in `maps` pool contain the map for each volume
- Has its own userspace tools: `vlmc` and `xseg`

Archipelago

Issues with Archipelago

- Debugging can be a serious pain
- Issues with `shm` can cause data corruption
- No easy way to map objects to a particular volume
- No garbage collection :)
- Synnefo has a ’hardcoded’ dependency on Archipelago, VM images are still stored using Archipelago. Will eventually be replaced (custom solution, Glance?).

RBD

RBD vs Archipelago

Since 2017, all our new ~okeanos clusters will use RBD instead of Archipelago. We took that decision for the following reasons:

- RBD is a part of Ceph
- RBD is stable
- Features that were missing from Archipelago
- Multiple ways to access images (krbd, librbd)
- Large community of users and devs behind it
- No need to maintain an in-house tool

krbd vs librbd. Which one?

Two things to evaluate: **Operational issues** and **performance**. Went along with **librbd**:

- Almost **same** performance (1 exception: seq read)
- krbd uses the **host's page cache**: can cause crashes, hung tasks
- librbd has **no kernel version dependency**
- librbd has a nice **admin socket**
- librbd's cache is easily configurable through ceph.conf
- No stale mapped devices with librbd

Issues:

- No monitoring yet on librbd (in progress)
- librbd opens up **a lot** of connections -> Evaluating async msg
- Had to write custom extstorage provider for Ganeti

Mega props to `alexaf` for his work on `librbd` vs `krbd`.

Ops

Automation & Config Management

We use extensively **Puppet** and **Python Fabric** all across our infra.

- Using puppet **only** for **configuration management**
- **No remote execution or Ceph provisioning**
- **In-house** modules for Ceph: Most modules out there perform deployment tasks
- Extensive use of **role/profile** pattern
- **Python Fabric** scripts for a lot of tasks: Deployments, upgrades, maintenance etc.
- **mcollective** for non-critical operations
- A large number of **Engineering Runbooks** for most operations.

Monitoring

We extensively monitor our Ceph infrastructure with a lot of tools:

- **icinga/checkmk**, mostly in-house stuff for host and ceph status
- **munin**, mainly for debugging purposes
- **collectd/graphite** for ceph metrics
- **prometheus** for networking and disk monitoring
- **ELK** for collecting, storing and analyzing host and ceph logs

Outages

Major outage on rd0

On *2016-09-09* we had our first multi-day Ceph outage

- OSDs of 1 node started flapping
- After 5 hours, flapping OSDs crashed
- 100 PGs peering+down, unfound objects (rc = 2 anyone?)
- MONs leaked FDs, lost communication
- Failed OSDs could not start with multiple assertions
- LevelDB corruption

Cluster recovered after 4 days of gdb sessions, the creation of an temp OSD, LevelDB repairing and more...

Root cause was never found: Hardware? Cosmic Radiation? Ghosts?

(Report on the outage on last slide)

Minor outages

- Nodes flapping due to faulty hardware
- Second node crashing while recovery is in progress
- Volumes mapped on two nodes
- Two same VMs running on different nodes…

Lessons learned

Lessons learned

- Never ever ever ever run a cluster with replica count = 2
- One crazy OSD/switch/NIC can ruin a whole cluster
- Ceph is hard :)
- Be up-to-date: a lot of bugfixes, improvements, features
- Docs are not always up-to-date, most info found in ML
- Benchmarking is difficult
- Deep-scrub can degrade performance, but is very important
- Test hardware thoroughly before running Ceph on it

Wishlist and TODO

Wishlist and TODO

- More tunings on ceph.conf and crushmap
- Try to keep up with new Ceph versions and features (bluestore, CephFS)
- Evaluate async messenger
- Evaluate RBD features (striping, snapshotting, layering etc)
- Better monitoring
    - Try prometheus exporter for ceph and experiment with alerting
    - Better ELK filtering
    - Maybe increase loglevels on some subsystems
    - Better hard disk monitoring
- Gain some experience on rgw, CephFS
- Gain some insight in Ceph code
- Some more automation would be nice
- ...

Links

https://grnet.gr

https://twitter.com/grnetnoc

https://www.synnefo.org/

https://www.synnefo.org/docs/archipelago/latest/

https://grnet.github.io/ganetimgr/

https://blog.noc.grnet.gr/2016/10/18/surviving-a-ceph-cluster-outage-the-hard-way/

https://www.terena.org/activities/tf-storage/ws18/slides/120215-archipelago.pdf

whoami

whoami

Nikos Kormpakis aka nkorb aka nkorbb

Systems & Services Engineer at GRNET, messing around with:

- IaaS platforms (ViMa, ~okeanos)
- Storage systems (Ceph, NetApp ONTAP boxes)
- Puppet
- a ton of other stuff (monitoring, multiple web stacks etc)

Mainly interested in storage, virtualization, OS internals, networking stuff, libre software, etc etc etc.

Find me at `nkorb_at_noc(dot)grnet(dot)gr` & `blog.nkorb.gr`