
CERN

Ceph in production

Hervé Rousseau (hroussea@cern.ch)
Data and Storage Services group | CERN IT Department

Outline

- Why Ceph ?
- Our setup and use cases
- Lessons learned
- Next steps

Why Ceph in addition to EOS ?

- Looking for
  - A block-storage for Openstack
  - Storage consolidation across AFS/NFS/etc…
- Candidates were
  - NetApp
  - GlusterFS
  - Ceph

Our production setup

46 machines (3 per Rack), each has
- 20 data disks (XFS)
- 4 SSDs (for journals)
- 10 Gbit/s Ethernet

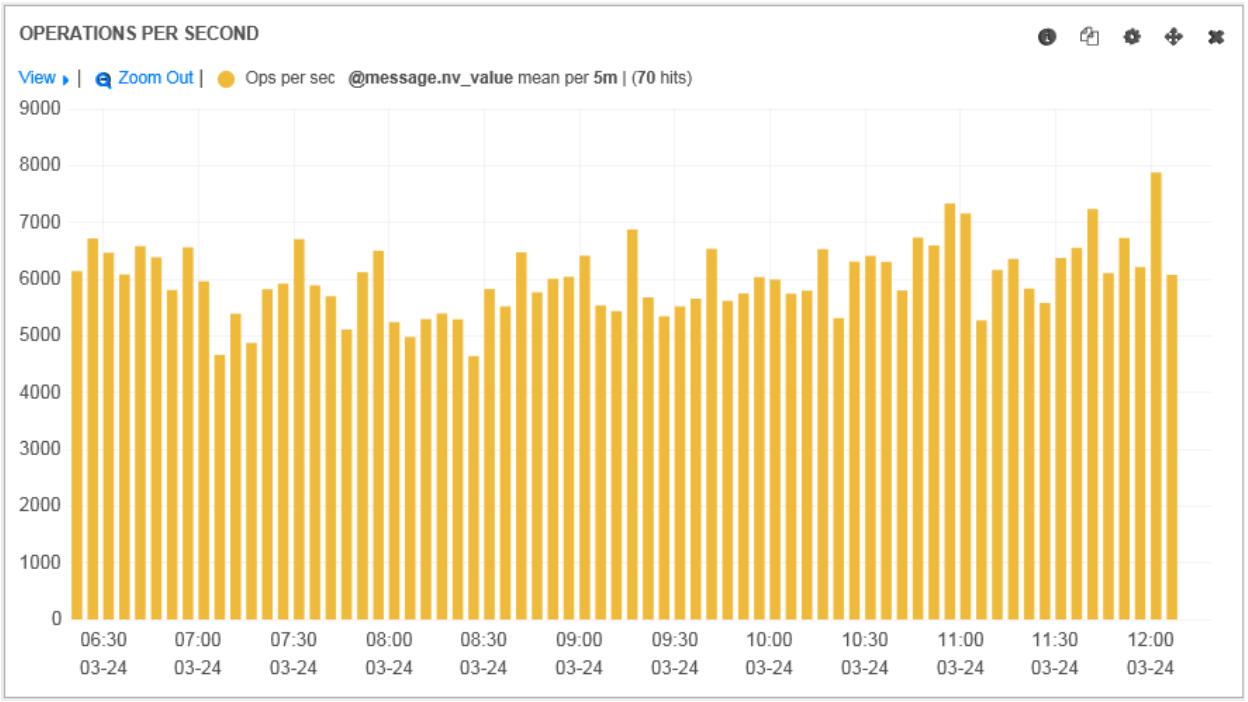
2.5 PB Raw ~ 850 TB Usable
(with 3 replicas)

http://cds.cern.ch/record/40386

Services provided

Storage backend for OpenStack Cinder

- 1.1k persistent storage volumes



Services provided

Virtual NFS Filer:

Replacement for NetApp filers for specific use cases

- Exposes Ceph RBD volumes to users via NFS
- ZFS for space management (and nice features: snapshots, differential replication)

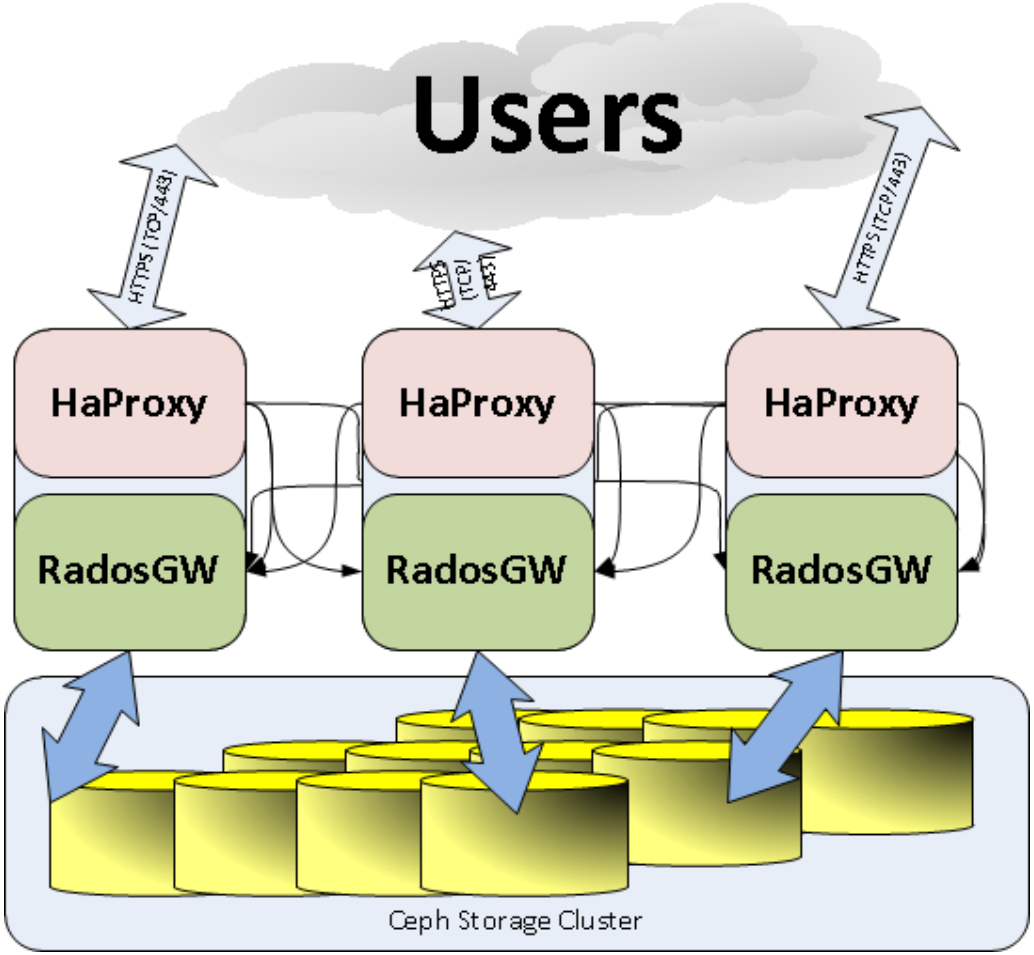
https://flic.kr/p/qMuRUx

Services ~~provided~~ testing

S3 gateway (RadosGW)

- "Standard" outside HEP
- Replaced FastCGI+Apache with Civetweb
- Haproxy for TLS and Load balancing
- Ran the CVMFS LHCb Test successfully (24 Million files / 1.4 TB)

Services provided



Lessons learned

- Dumpling to Firefly upgrade

Alice in Wonderland – Disney 1951

Lessons learned

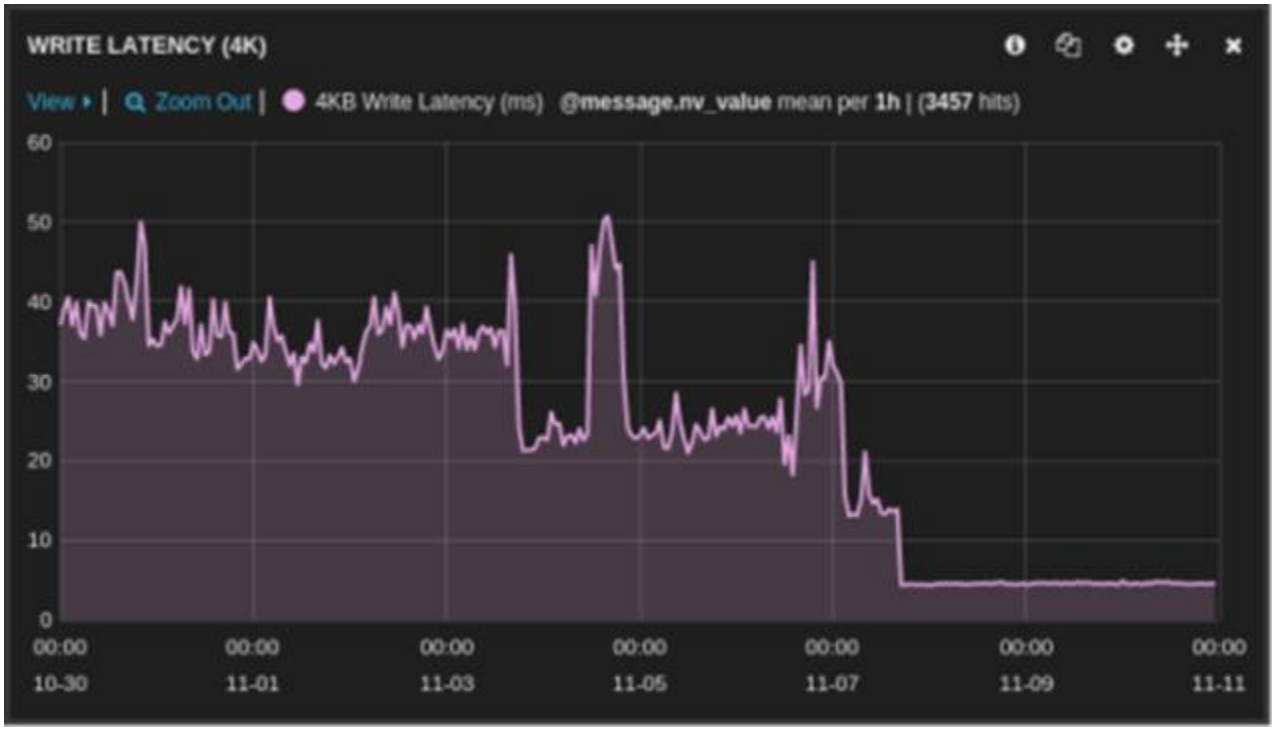
- Dumpling to Firefly upgrade
  - Careful planning
  - Read the Release notes
  - Usually upgrade the MONs, then the OSDs and finally the MDS and RadosGWs

Lessons learned

- Write latency
- Each write is first committed to the journals
- By default the journal is stored next to the data, which results in less-than-optimal IOPS numbers.
- Move the journal to dedicated partitions on SSDs

Lessons learned

- Write latency



https://flic.kr/p/8ee93Z

Lessons learned

- Power outage on 16th Oct 2014
  - 3 out of 5 monitors went down
  - No quorum
  - No IO until quorum restored (for ~23 minutes)

Lessons learned

https://flic.kr/p/pjGY3Y

Lessons learned

- Data placement
  - Failure domains (Electrical, Network, etc…)
  - Number of PGs per OSD
- Many small objects and zone_reclaim_mode
- Read the documentation

Next steps (R&D)

Store data + namespace in Ceph

RadosFS: https://github.com/cern-eos/radosfs

EOS-Diamond: https://github.com/cern-eos/eos-diamond

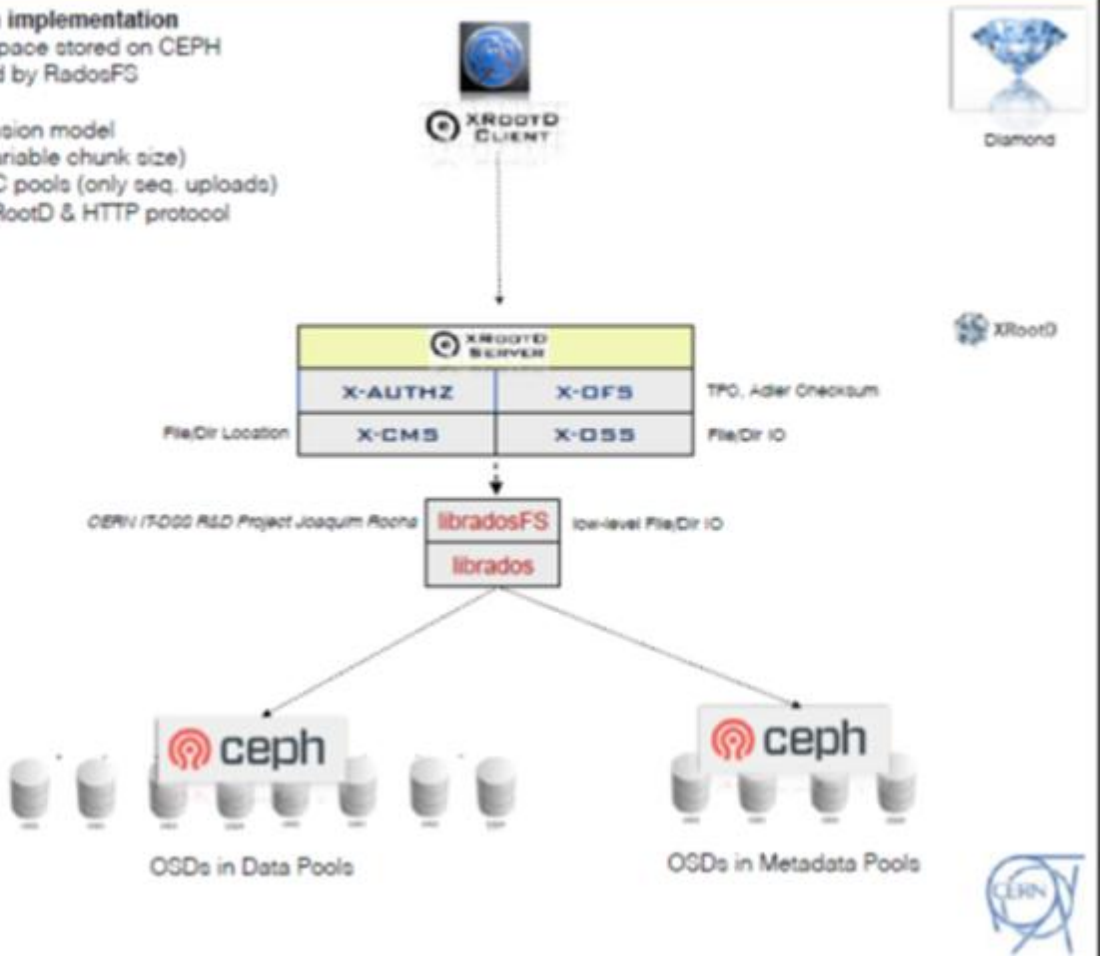
CERN
IT Department

EOS - Diamond R&D - 1st generation

CERN IT Department

1st generation implementation
- files & namespace stored on CEPH
  - implemented by RadosFS

- POSIX permission model
- parallel IO (variable chunk size)
- support for EC pools (only seq. uploads)
- tested with XRootD & HTTP protocol



Next steps

Ceph as a Castor backend

- Decouple disks and disk servers (diskservers become proxy)
- Avoids hotspots with multiple clients accessing same file

RadosStriper contributed by S. Ponce

Next steps

- Scale test
  - 150 machines
  - ~7300 OSDs
  - 25 PB

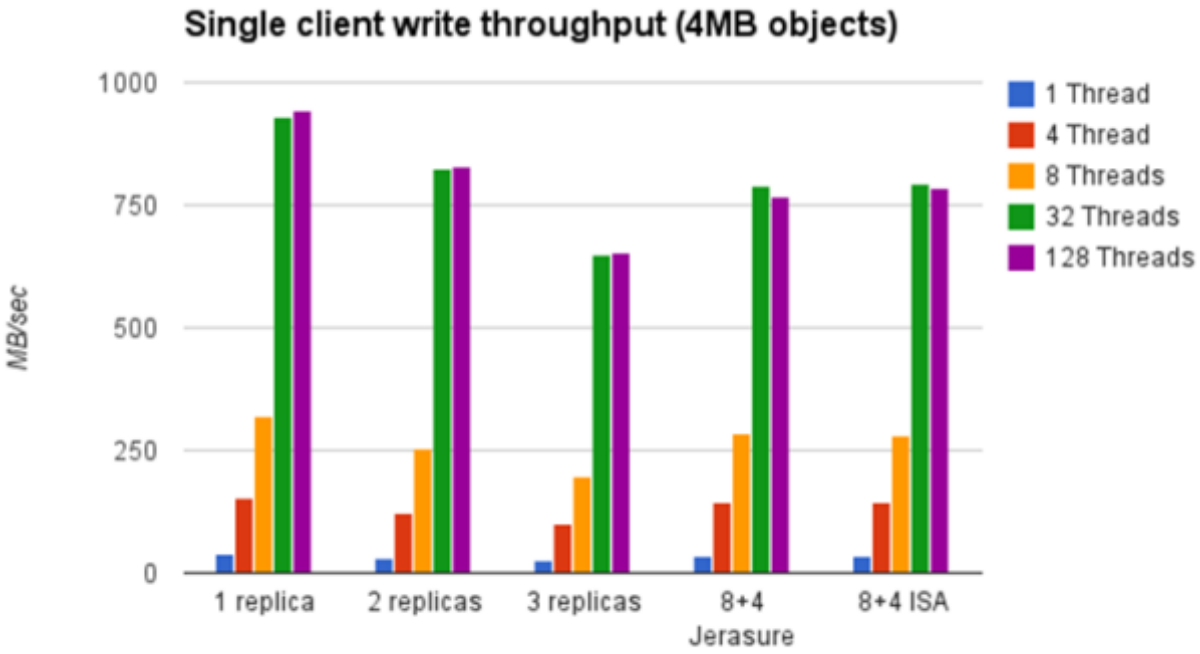
Tests done with Hammer (0.93-102)

Next steps

- Does it scale ?
  - Adding the nodes
  - OSD Memory consumption
  - MONs fitted with SSDs

Next steps

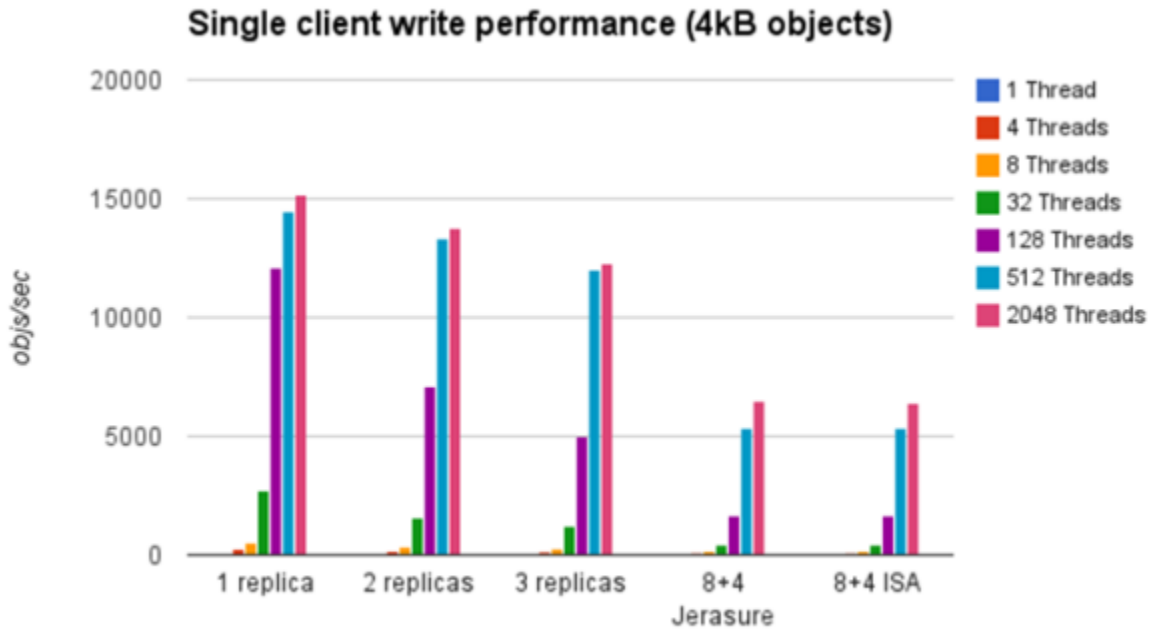
- Single client performance



ISA Support contributed by A. Peters

Next steps

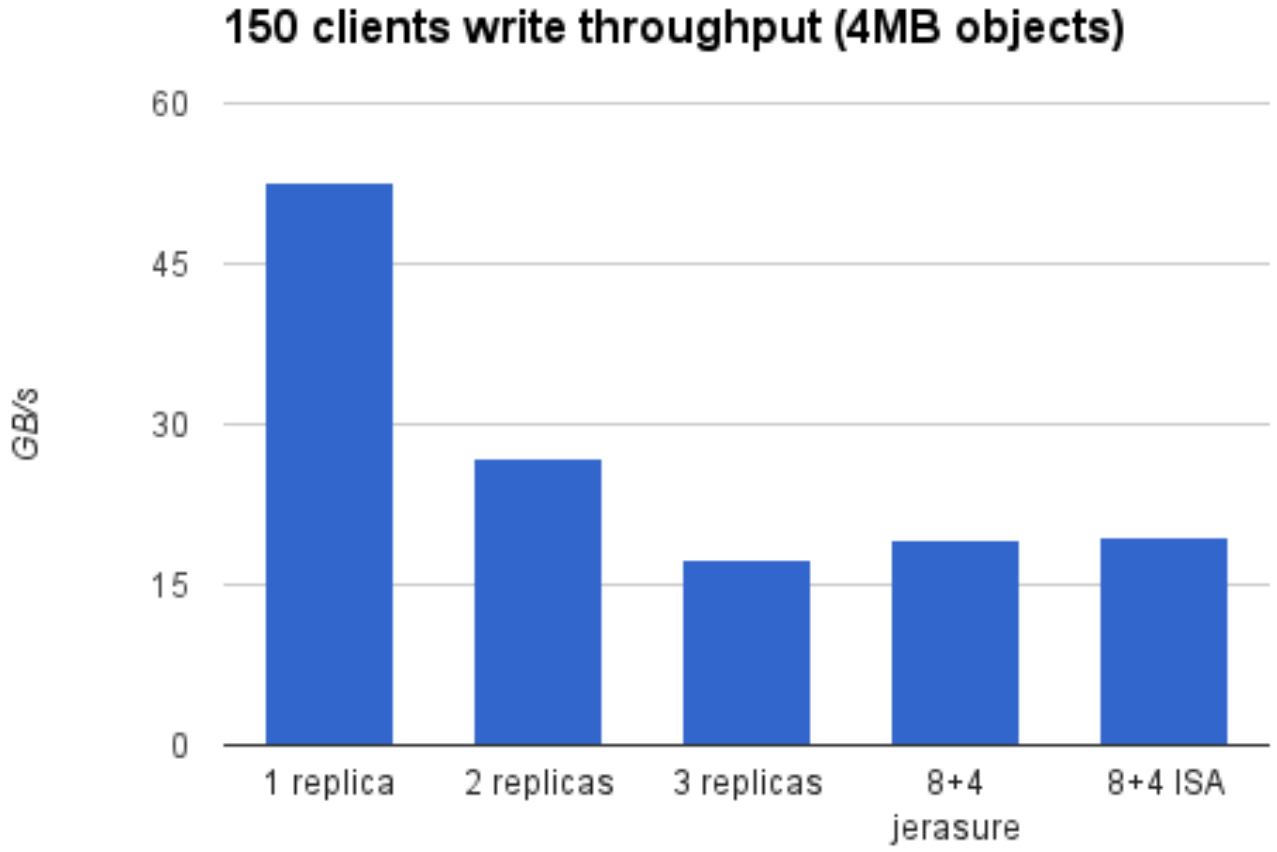
- Single client performance



ISA Support contributed by A. Peters

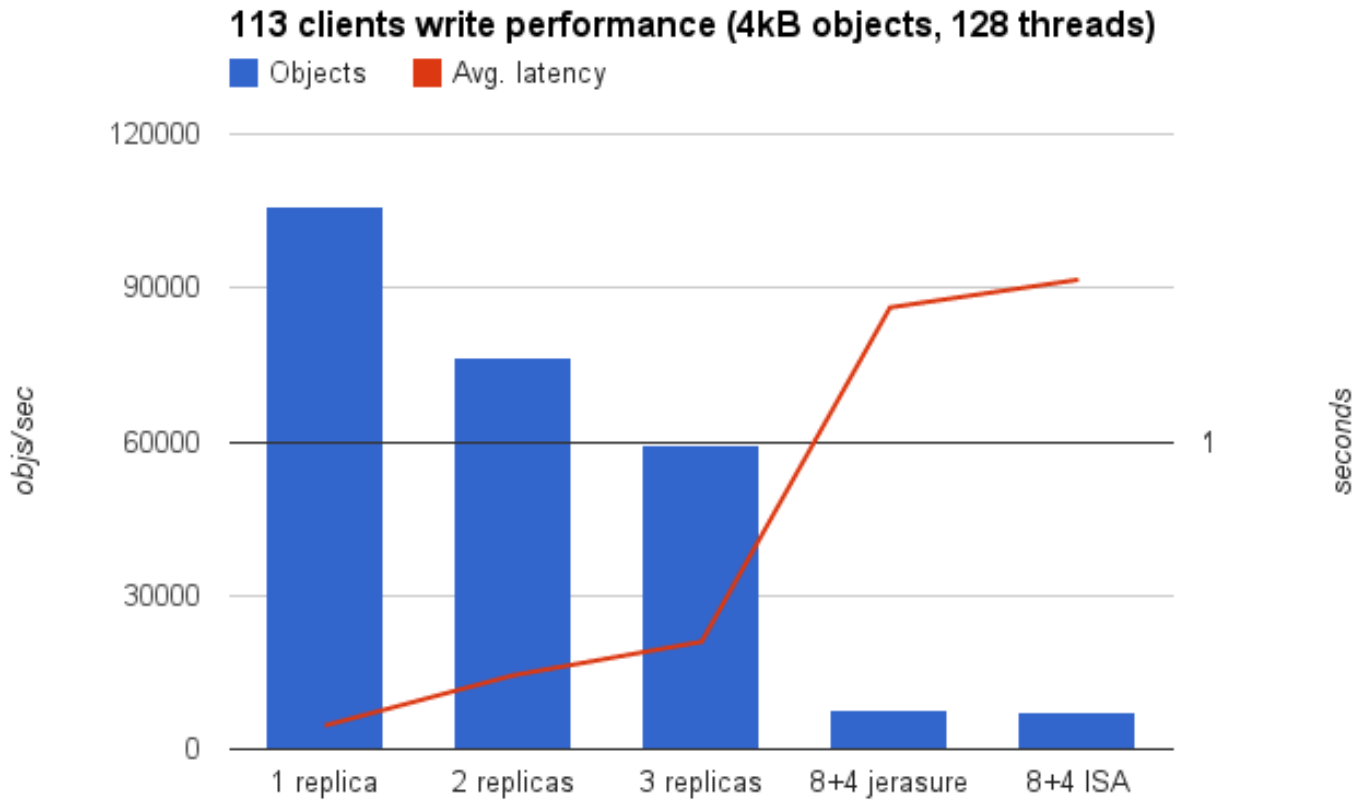
Next steps

- Large scale performance



Next steps

- Large scale performance



Conclusion

- “Openstack” like usage: running smoothly for over 1 year
- Large scale O(10k) cluster: not straightforward, some work ahead
- Ceph community is great and very responsive

www.cern.ch

Additional informations

Glossary

- MON: Monitor (Administrative, no data)
- OSD: Object Storage Daemon (stores data)
- MDS: Metadata Server (for CephFS)
- RADOS: The core set of storage software which stores the user's data (MON+OSD).
- Rados GW: S3 Gateway to RADOS
- RBD: Block storage on top of RADOS

(etc...)

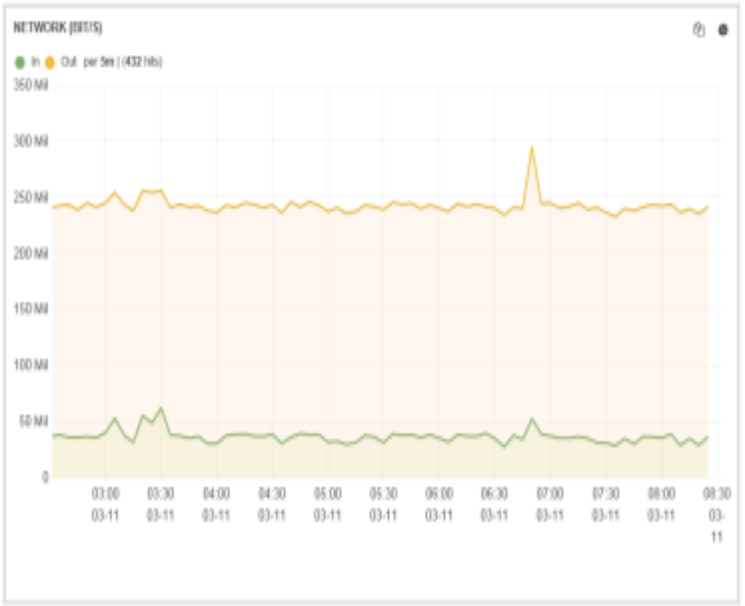
http://ceph.com/docs/master/glossary/

Large scale test: ceph.conf

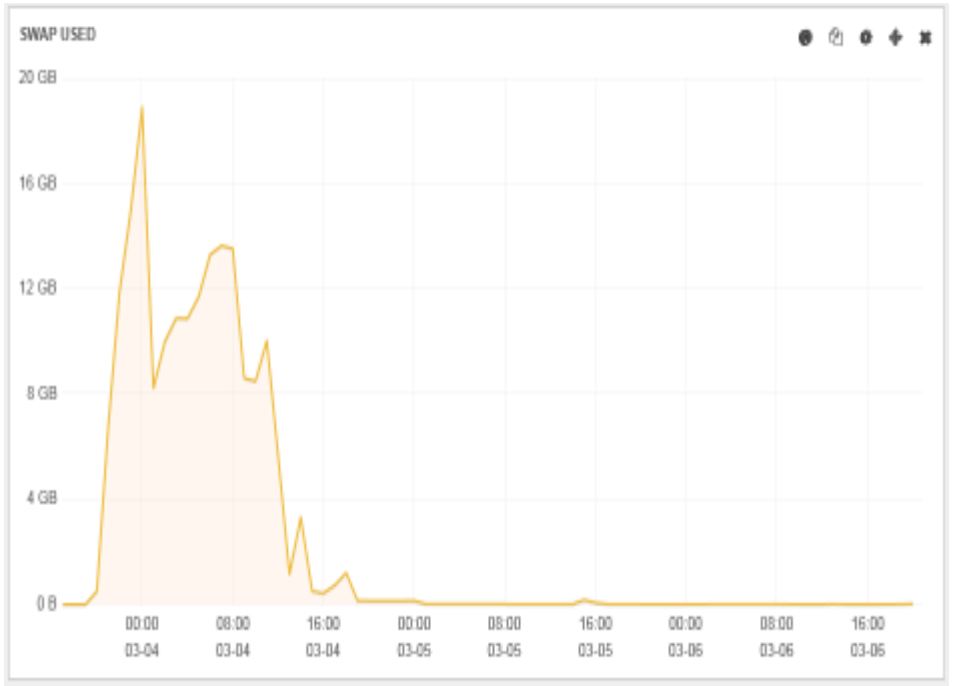
```
[global]
…
  osd map message max = 10 # Send that many maps per message

[mon]
…
  mon pg warn min per osd = 15
  mon pg warn max object skew = 20
```

```
[osd]
  # See http://www.spinics.net/lists/ceph-devel/index.html#23048
  osd map cache size = 20 # Not size but numbers of maps
  osd map max advance = 10
  osd map share max epochs = 10
  osd pg epoch persisted max stale = 10
```



Traffic to the MONs with idle cluster



Avg swap usage before tuning map parameters

Large scale test: pools and rules

```
pool 85 'testbb3' replicated size 3 min_size 1 crush_ruleset 5 object_hash rjenkins pg_num 4096 pgp_num
4096 last_change 84749 flags hashpspool stripe_width 0
pool 86 'testbb2' replicated size 2 min_size 1 crush_ruleset 5 object_hash rjenkins pg_num 4096 pgp_num
4096 last_change 84753 flags hashpspool stripe_width 0
pool 92 'testbb1' replicated size 1 min_size 1 crush_ruleset 5 object_hash rjenkins pg_num 4096 pgp_num
4096 last_change 84786 flags hashpspool stripe_width 0
pool 93 'testbbec-isa' erasure size 12 min_size 8 crush_ruleset 12 object_hash rjenkins pg_num 4096
pgp_num 4096 last_change 84789 flags hashpspool stripe_width 4096
pool 94 'testbbec-jerasure' erasure size 12 min_size 8 crush_ruleset 13 object_hash rjenkins pg_num
4096 pgp_num 4096 last_change 84793 flags hashpspool stripe_width 4096
```

- Crush rule

```
rule bigbang {
        ruleset 5
        type replicated
        min_size 1
        max_size 10
        step take bigbang
        step chooseleaf firstn 0 type
rack
        step emit
}
```

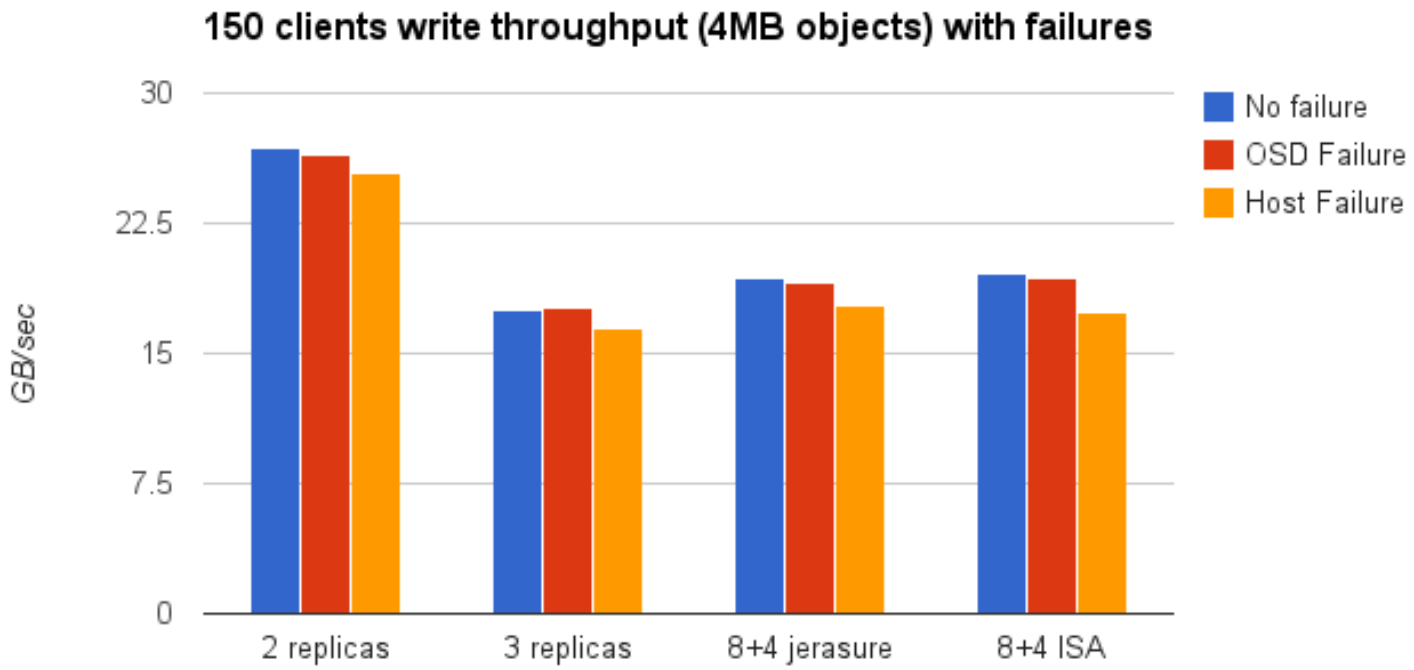
- Erasure Code Profile:

```
k=8
m=4
plugin=jerasure
ruleset-failure-domain=host
ruleset-root=bigbang
technique=reed_sol_van
```

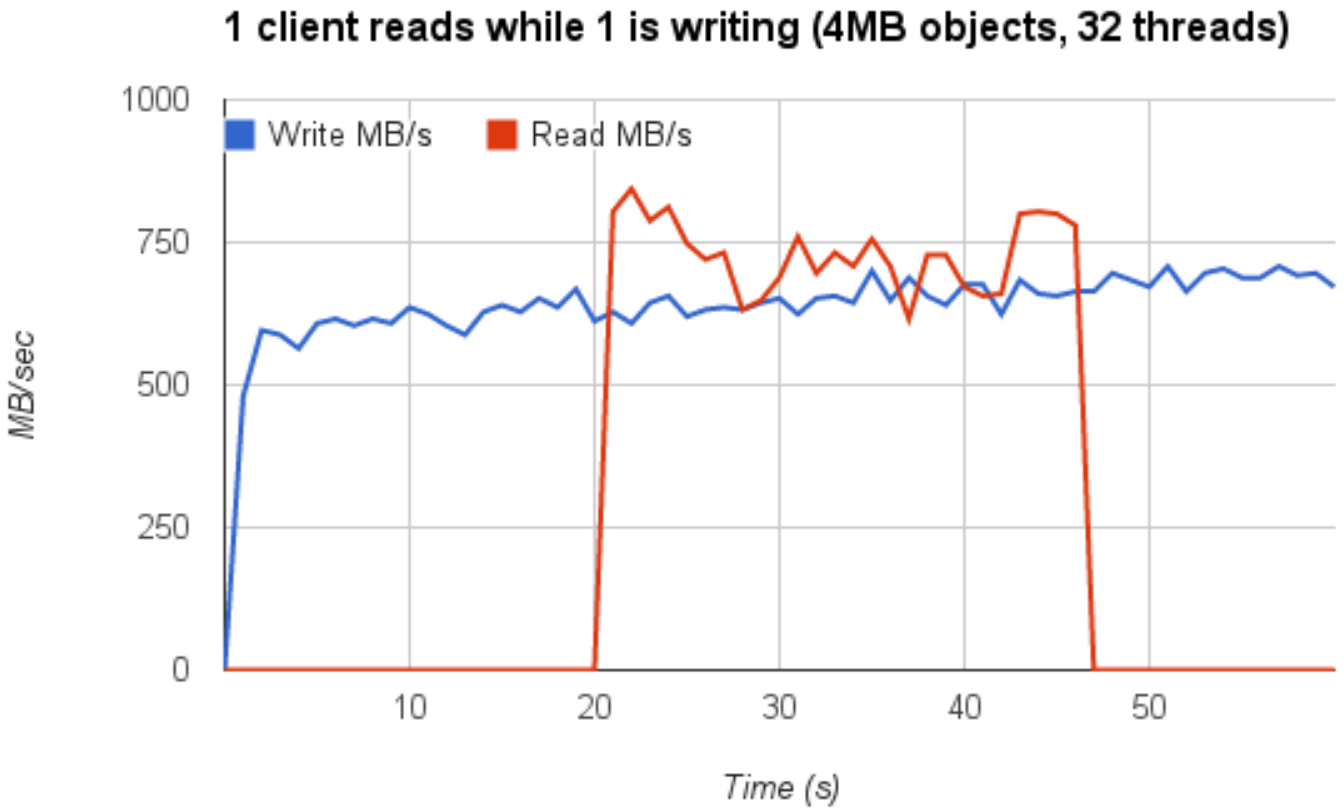
Large scale test: commands

```
for s in 4096 4194304; do
  for p in testbb1 testbb2 testbb3 testbbec-jerasure testbbec-isa; do
    for i in 1 4 8 32 128 512 2048; do
      echo '* * ' $p $i
      echo rados bench -p $p 10 write -b $s -t $i --no-cleanup
      rados bench -p $p 10 write -b $s -t $i --no-cleanup
      echo rados cleanup -p $p
      time rados cleanup -p $p
    done
  done
done
```

Large scale test: Misc. charts



Large scale test: Misc. charts



On a 3 replicas pool

RadosFS Features

CephFS vs. RadosFS

CERN IT Department

All inodes inside directory (meta data pool) File open via MDS cached directory - access control on directory object	File inodes in data pool, directory inodes in directory (meta data pool) File open via Inode lookup in Data Pool - access control on file object
Directories stored as 2-level log structured storage	Directories stored as change log
Single active MDS, multiple passive failover MDS - scalable by subtree - *unstable code* for multiple active MDS	Every OSD contributes to MDS - scale by directory/OSDs
MDS service is remote to directory object	MDS service is local to directory object
Extended attributes on file & directory object	Extended attribute on file & directory object and directory entries
full POSIX, very advanced kernel client with FScache implementation	reduced POSIX, no kernel client envisaged
parallel IO with stripe unit & size	parallel IO with stripe unit & size
large community interest as generic filesystem	<can not answer>
chaotic code base, external, complicated	clean code base, in house, simple
based on trusted clients	any kind of desired security upstream


Diamond

Large scale test: CephFS

Works but we didn't focus on it during our tests:

- Mandatory df output:

```
# df -h | grep ceph
ceph.cern.ch:6789:   26P  927T   25P   4% /cephfs
```

✓ Inline data