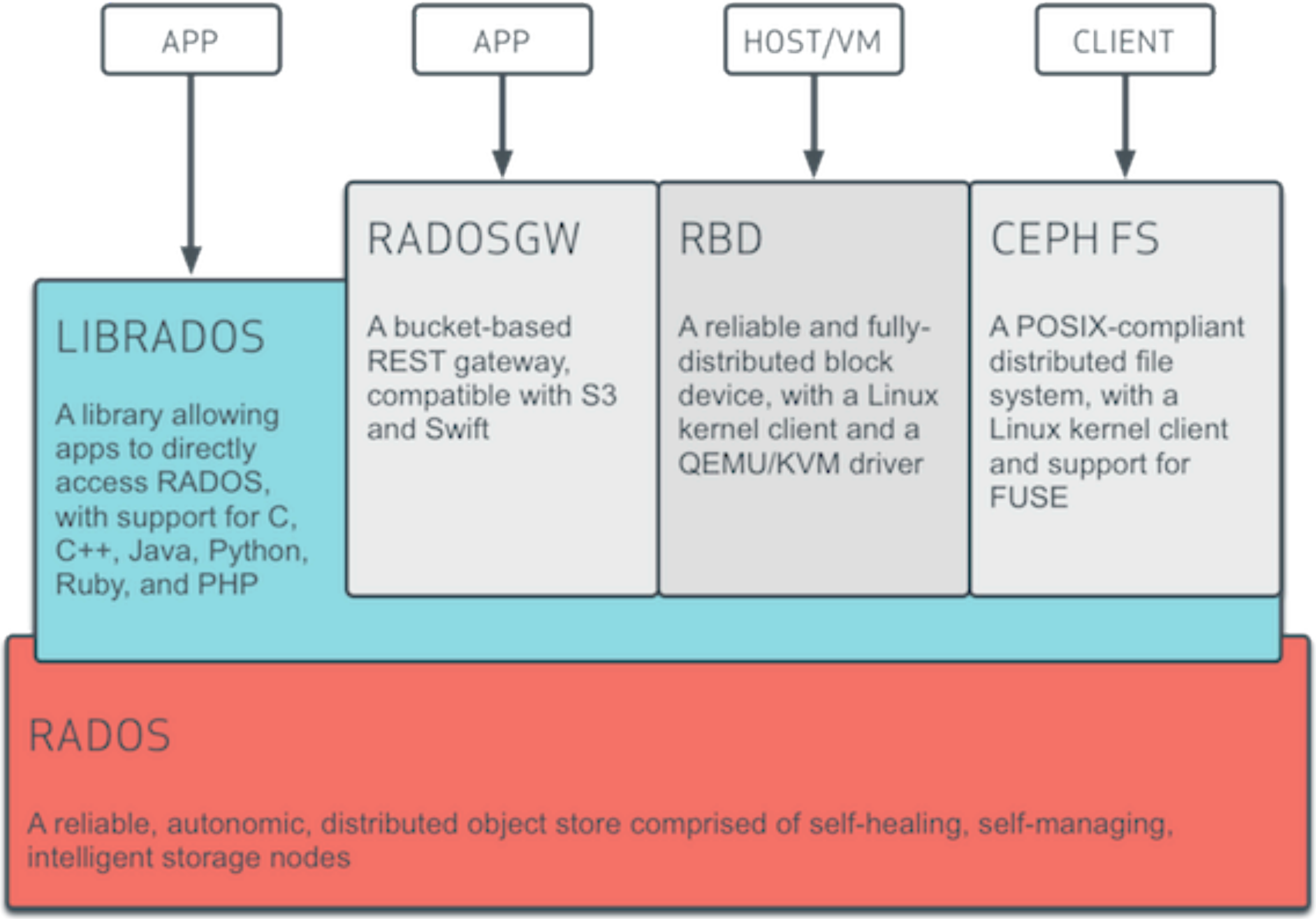
Ceph @ CERN: one year on…

Dan van der Ster (daniel.vanderster@cern.ch)
Data and Storage Service Group | CERN IT Department

HEPIX 2014 @ LAPP, Annecy

Ceph Architecture and Use-Cases



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

OpenStack + Ceph

- Used for *Glance* Images, *Cinder* Volumes and *Nova* ephemeral disk (coming soon)

- Ceph + OpenStack offers compelling features:
    - CoW clones, layered volumes, snapshots, boot from volume, live migration
    - Cost effective with Thin Provisioning
        - ~110TB “used”, ~45TB * replicas on disk

- Ceph is the most popular network block storage backend for OpenStack
    - http://opensource.com/business/14/5/openstack-user-survey

Ceph at CERN

- In January 2013 we started to investigate Ceph for two main use-cases:
  - Block storage for OpenStack
    - Other options being NetApp (expensive, lock-in) and GlusterFS
  - Storage consolidation for AFS/NFS/…

- We built a 250TB test cluster out of old CASTOR boxes, and early testing was successful so we requested hardware for a larger prototype…

3PB of Ceph

47 disk servers/1128 OSDs

Dual Intel Xeon E5-2650

32 threads incl. HT

Dual 10Gig-E NICs

Only one connected

24x 3TB Hitachi disks

Eco drive, ~5900 RPM

3x 2TB Hitachi system disks

Triple mirror

64GB RAM

5 monitors

Dual Intel Xeon L5640

24 threads incl. HT

Dual 1Gig-E NICs

Only one connected

2x 2TB Hitachi system disks

RAID-1 mirror

1x 240GB OCZ Deneva 2

/var/lib/ceph/mon

48GB RAM

```
# df -h /mnt/ceph
Filesystem
Size   Used Avail Use% Mounted on
xxx:6789:/  3.1P  173T  2.9P   6% /mnt/ceph
```

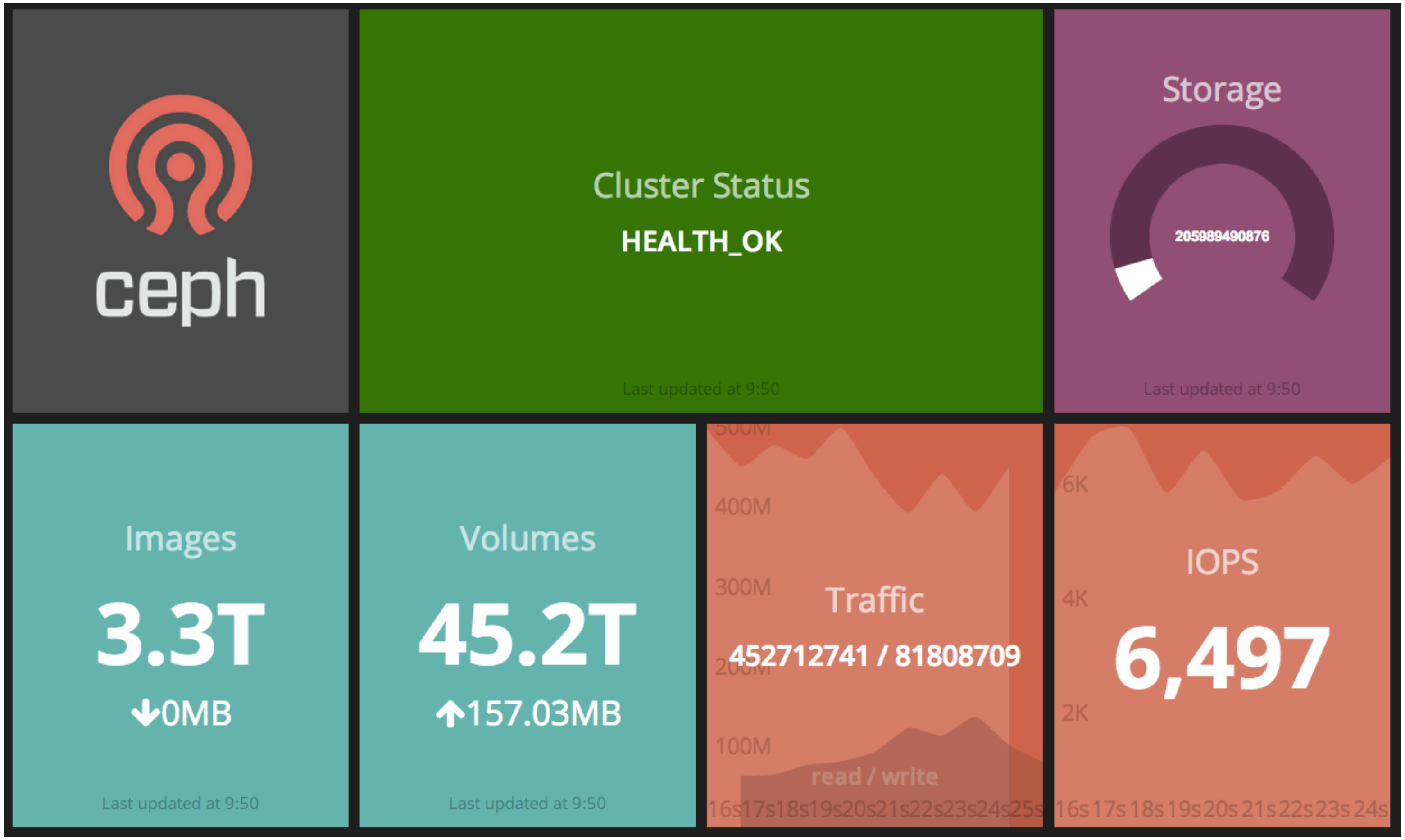
Deployment

- Fully puppetized using forked upstream module: https://github.com/cernceph/puppet-ceph

- Automated machine commissioning and maintenance
  - Add a server to the hostgroup (osd, mon, radosgw)
  - OSD disks are detected, formatted, prepared, auth'd
    - Also after disk replacement
  - Auto-generated ceph.conf
  - Last step is manual/controlled: service ceph start

- Mcollective for bulk operations on the servers
  - Ceph rpm upgrades
  - daemon restarts

A “Dashing” dashboard



Cluster Status

HEALTH_OK

Last updated at 9:50

Storage

205989490876

Last updated at 9:50

Images

3.3T

↓0MB

Last updated at 9:50

Volumes

45.2T

↑157.03MB

Last updated at 9:50

Traffic

452712741 / 81808709

read / write

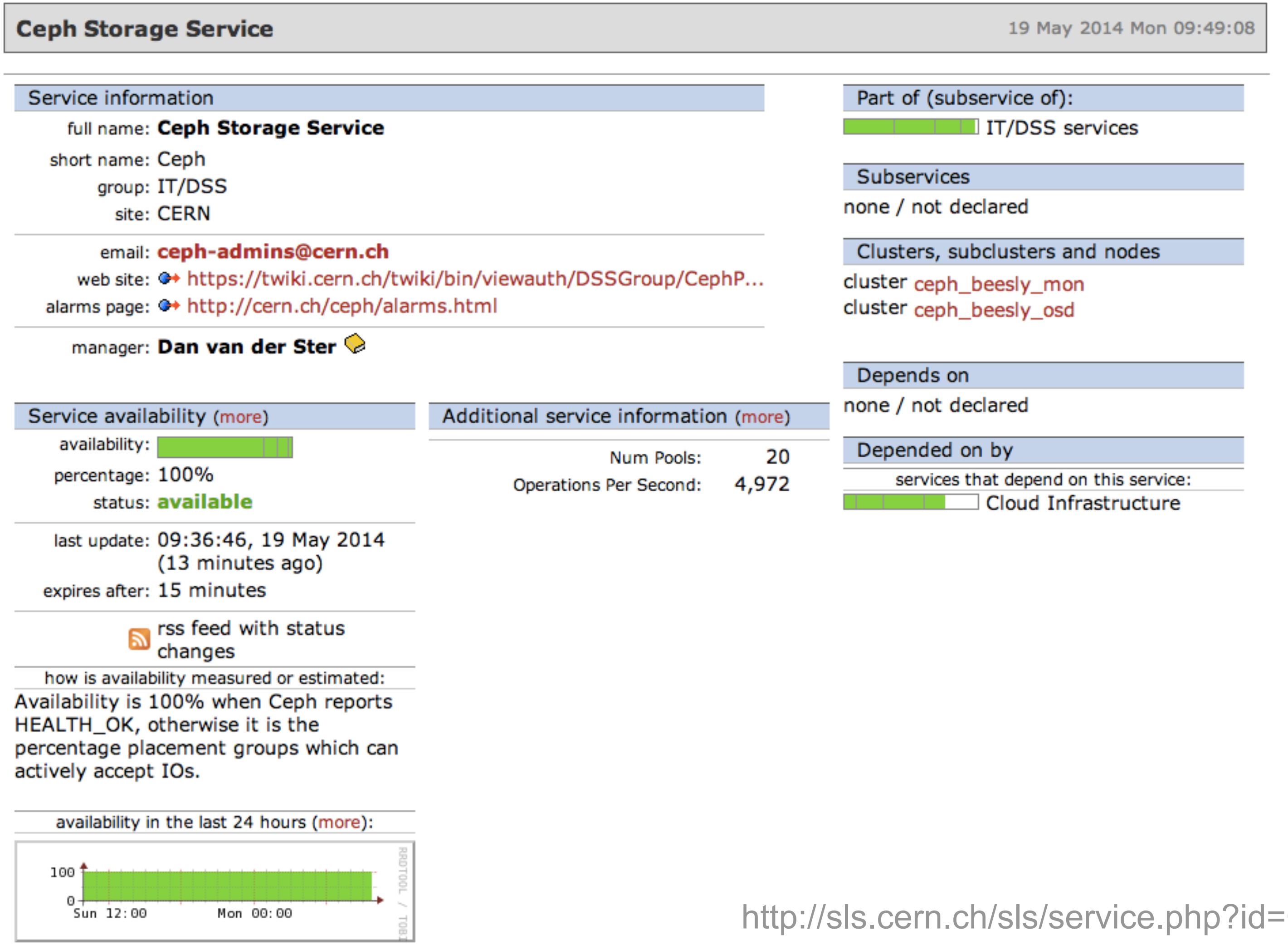
IOPS

6,497

ceph
500M
400M
300M
200M
100M
16s17s18s19s20s21s22s23s24s25s
6K
4K
2K
16s 17s 18s 19s 20s 21s 22s 23s 24s

Code: https://github.com/rochaporto/dashing-ceph

SLS Monitoring



Ceph Storage Service 19 May 2014 Mon 09:49:08

Service information

full name: **Ceph Storage Service**
short name: Ceph
group: IT/DSS
site: CERN

email: **ceph-admins@cern.ch**
web site: https://twiki.cern.ch/twiki/bin/viewauth/DSSGroup/CephP...
alarms page: http://cern.ch/ceph/alarms.html

manager: **Dan van der Ster**

Service availability (more)

availability: 🟩
percentage: 100%
status: **available**

last update: 09:36:46, 19 May 2014 (13 minutes ago)
expires after: 15 minutes

rss feed with status changes

how is availability measured or estimated:

Availability is 100% when Ceph reports HEALTH_OK, otherwise it is the percentage placement groups which can actively accept IOs.

availability in the last 24 hours (more):

Additional service information (more)

Num Pools:	20
Operations Per Second:	4,972

Part of (subservice of):

🟩 IT/DSS services

Subservices

none / not declared

Clusters, subclusters and nodes

cluster ceph_beesly_mon
cluster ceph_beesly_osd

Depends on

none / not declared

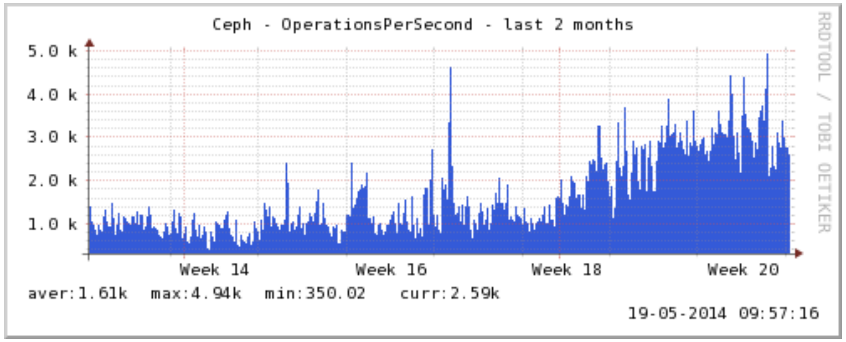
Depended on by

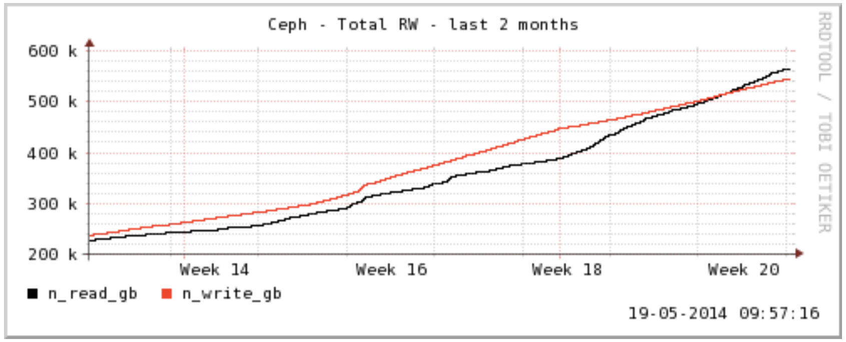
services that depend on this service:

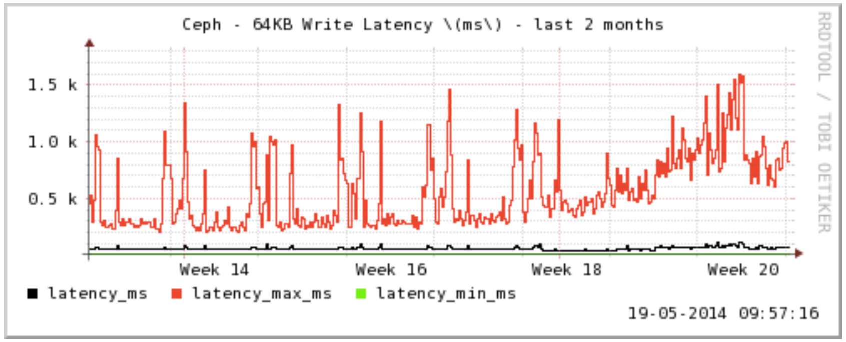
🟩 Cloud Infrastructure

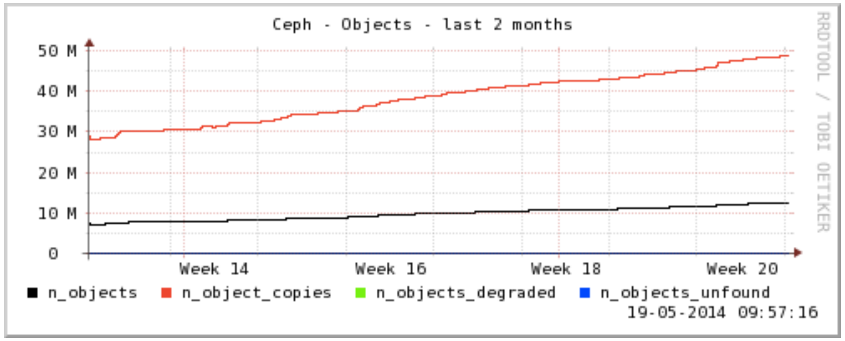
http://sls.cern.ch/sls/service.php?id=Ceph

Example SLS plots

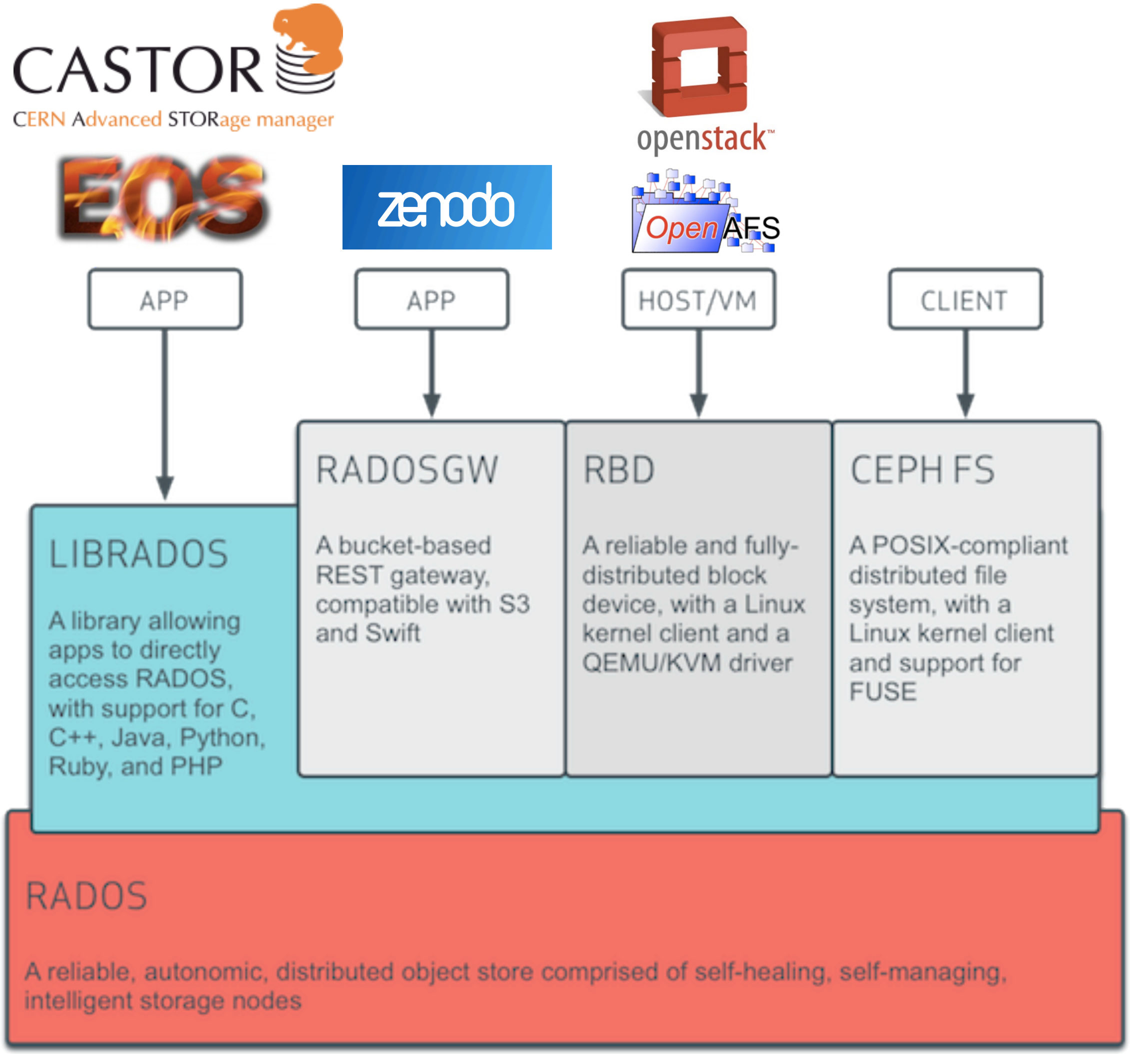








Potential Use-Cases



LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Ceph for Physics Data?

- RADOS is not a drop-in HEP storage system
  - No namespace
  - Object size limitations
  - No X509/kerberos
  - Much more …
- (EOS/Dcache/DPM/...) on RBD would allow thin disk servers, but they still act as “gateways” to the data on Ceph
  - double/triple/quadruple network traffic
- CephFS is NFS-like, but it lacks strong auth (among other things). See our dev blueprint: http://wiki.ceph.com/Planning/Blueprints/Firefly/Strong_AuthN_and_AuthZ_for_CephFS

CASTOR & XRootD/EOS

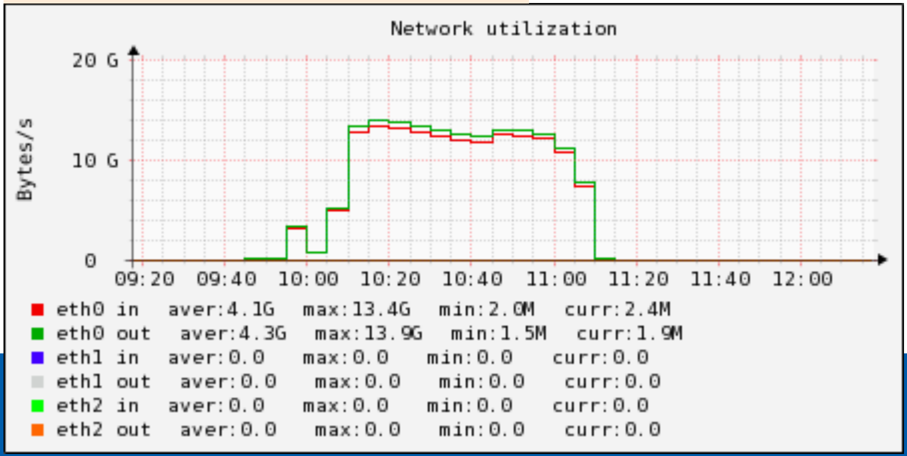
- Exploring RADOS backend for these storage systems
- CASTOR needs raw throughput performance (to feed many tape drives at 250MBps each).
  - Striped RWs across many OSDs are important.
  - Rados Striper for CASTOR: https://github.com/ceph/ceph/pull/1186
- XRootD/EOS may benefit from the lack of a namespace to store O(billion) objects
  - Bonus: also http/webdav with X509/kerberos, possibly even fuse mountable.
  - RADOS FS: https://github.com/joaquimrocha/radosfs
  - Xrootd Plugin: https://github.com/joaquimrocha/xrootd-rados-oss
- Developments are exploratory / early stages.

Throughput testing

basic rados bench - *saturate the network*

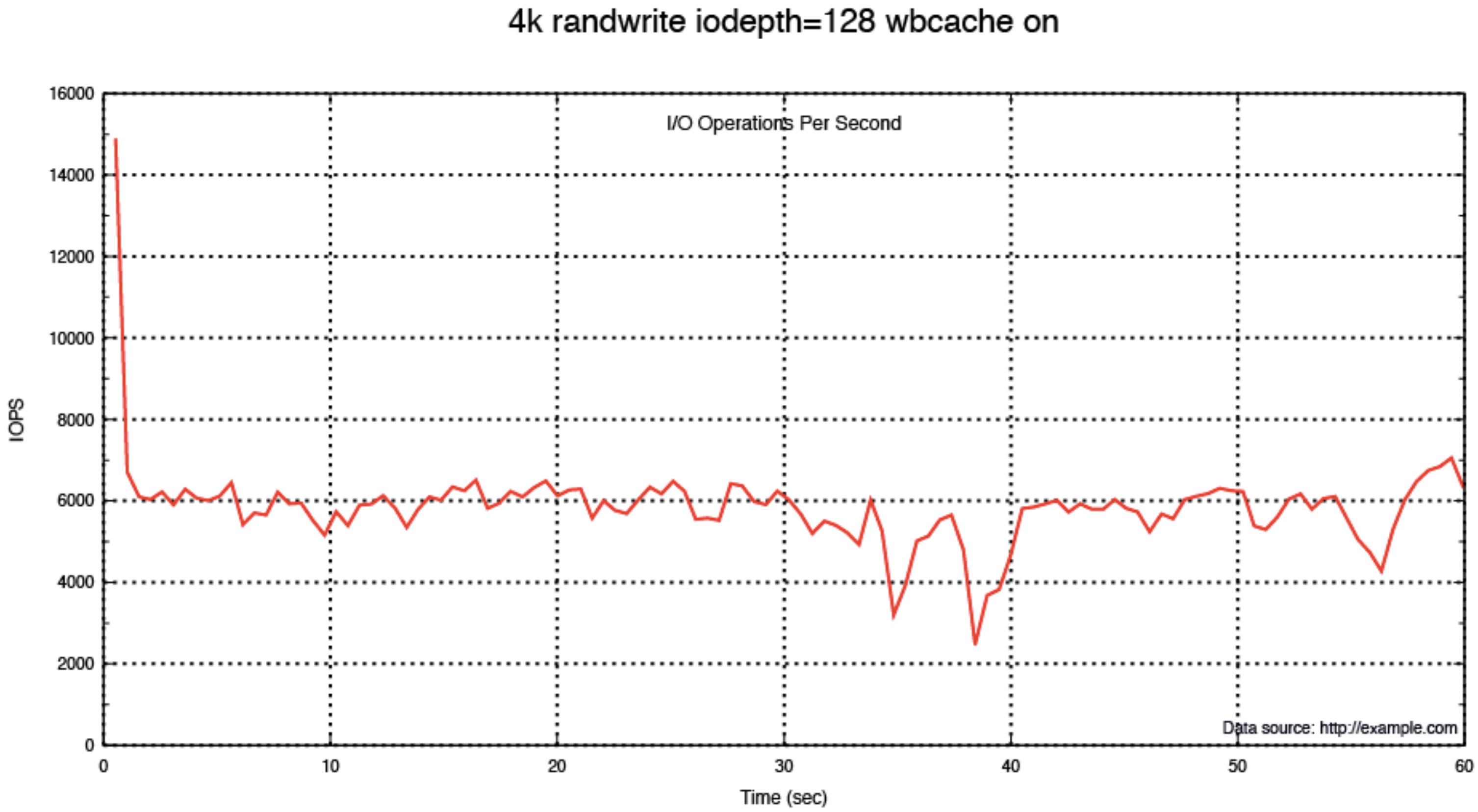
```
[root@p05151113471870 ~]# rados bench 30 -p test write -t 100
Total writes made:      7596
Write size:             4194304
Bandwidth (MB/sec):     997.560
Average Latency:        0.395118
[root@p05151113471870 ~]# rados bench 30 -p test seq -t 100
Total reads made:     7312
Read size:            4194304
Bandwidth (MB/sec):    962.649
Average Latency:       0.411129
```

all-to-all rados bench



Striping across many objects gives high throughput performance

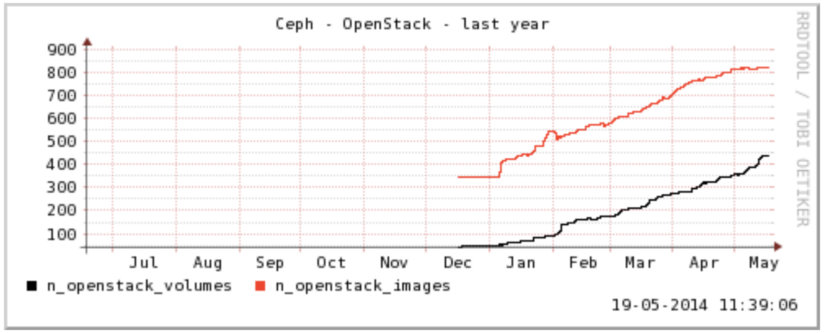
(Single-client) IOPS testing



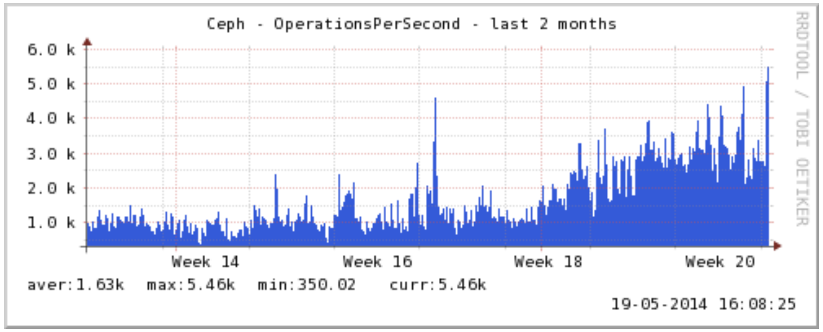
- VM: ~6000 4k randwrite iops to RBD vs ~100 iops on the local disk
- Total cluster capacity is ~20-30K iops (so we throttle the clients)

OpenStack Volumes & Images

- Glance: in production for >6 months
  - Only issue was to increase ulimit nofiles
- Cinder: in production since March.
  - ~400 volumes: >100TB allocated, ~45TB used, ~200TB including replicas



Growing # of volumes/images



Increasing IOPS, usually 5-6k now

Cinder Monitoring

- Throttling: OpenStack and qemu-kvm can throttle the block devices.
  - We use 400 iops_r, 200 iops_w, 80 mbps_w, 40 mbps_r
  - But this is probably too generous (Amazon EBS provides 100 IOPS)
  - We will scale this back soon to allow more users

- Latency: best case synchronous 64k write was 30-40ms
- With increased usage a 64k write can approach/exceed 100ms

- We log all IOs for analysis, for example on 8 May 2014:
  - 322,001,158 writes; 170,753,949 reads
  - 25% of writes were to the top four volumes.
  - 191,809,175 (74%) writes were 4kB.
  - 28% of reads were 512kB, 25% of reads were 4kB.

Why such high latency?

- Ceph writes *synchronously* to its OSD journal and asynchronously to the OSD filestore

Everything is written twice

Deployment question: Shared vs. dedicated journal devices

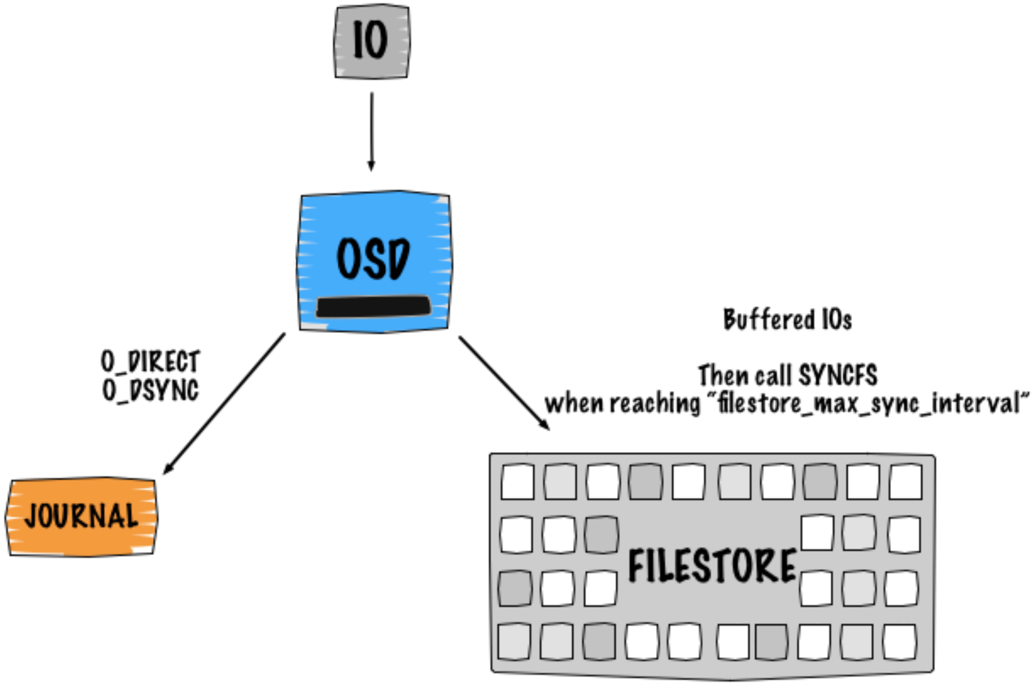


Image from http://www.sebastien-han.fr/

IOPS limitations

- Our config with spinning, co-located journals limit the servers to around 500 IOPS each
  - We are currently at ~30% of the total cluster IOPS
  - (and need to save room for failure recovery)
- Using SSDs journals (1 SSD for 5 disks) can at least double the IOPS capacity, and our tests show ~3x-5x burst IOPS

Scalability

- O(1000) OSDs seems to be doable
  - What about 10,000 or 100,000 OSDs?
  - What about 10,000 or 100,000 clients?
  - Many Ceph instances is always an option, but not ideal

- OSDs are scalable:
  - communicate with peers only (~100, no matter how large the cluster)

- Client process/socket limitations:
  - short lived clients only talk to a few OSDs – no scalability limit
  - Long lived clients (e.g. qemu-kvm) eventually talk to all OSDs – each with 1-2 sockets, ~2 processes.
    - Ceph will need to optimize for this use case in future (e.g. using thread pools…)

Other topics, no time

- 250 million objects test: 7 hours to backfill one failed OSD
- LevelDB troubles:
  - high cpu usage on a couple OSDs, had to scrap them
  - mon leveldb's grow ~10GB per week (should be 700MB)
- Backup: async geo-replication
- Object reliability: 2, 3 or 4 replicas; use the rados reliability calculator
- Slow requests: tuning the deadline elevator, disabling updatedb
- Don't give a cephx keyring to untrusted users: they can DOS your mon and do other untold damage
- Data distribution: CRUSH often doesn't lead to perfectly uniform data distribution. Use "reweight-by-utilization" to flatten it out.
- New "firefly" features to test: erasure coding, tiered pools
- RedHat acquisition: puts the company on solid footing, will they try to marry GlusterFS+Ceph?

Summary

- The CERN IT infrastructure is undergoing a private cloud revolution, and Ceph is providing the underlying storage.

- In nine months with a 3PB cluster, we’ve not had any disasters, and performance is at the limit of our hardware
  - For block storage, make sure you have SSD journals

- Beyond the OpenStack use-case, we have a few obvious and a few more speculative options: AFS, NFS, …, physics data

- Still young, still a lot to learn, but seems promising.

www.cern.ch