WHAT'S NEW IN JEWEL AND BEYOND

SAGE WEIL

CEPH DAY CERN – 2016.06.14

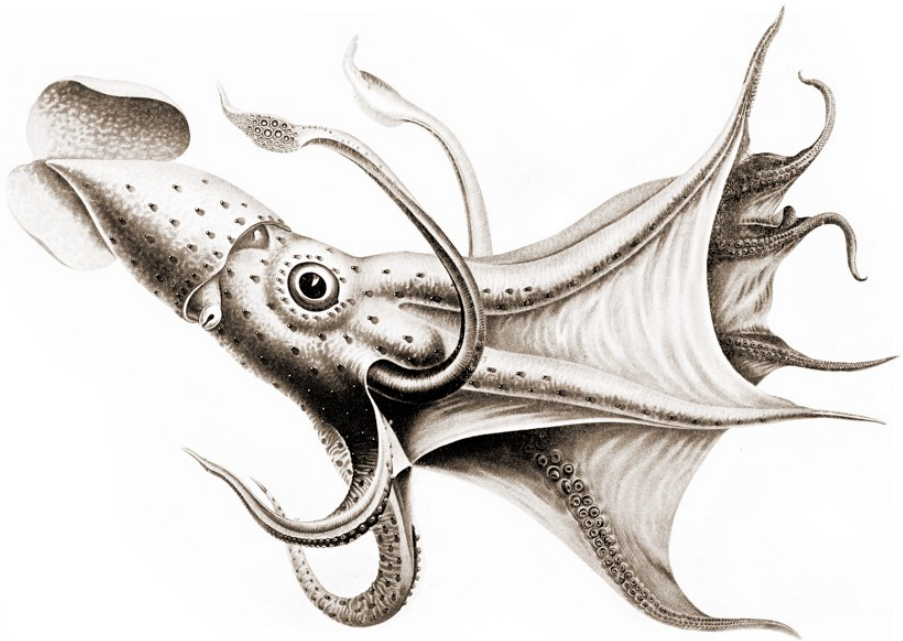
AGENDA

- New in Jewel
- BlueStore
- Kraken and Luminous

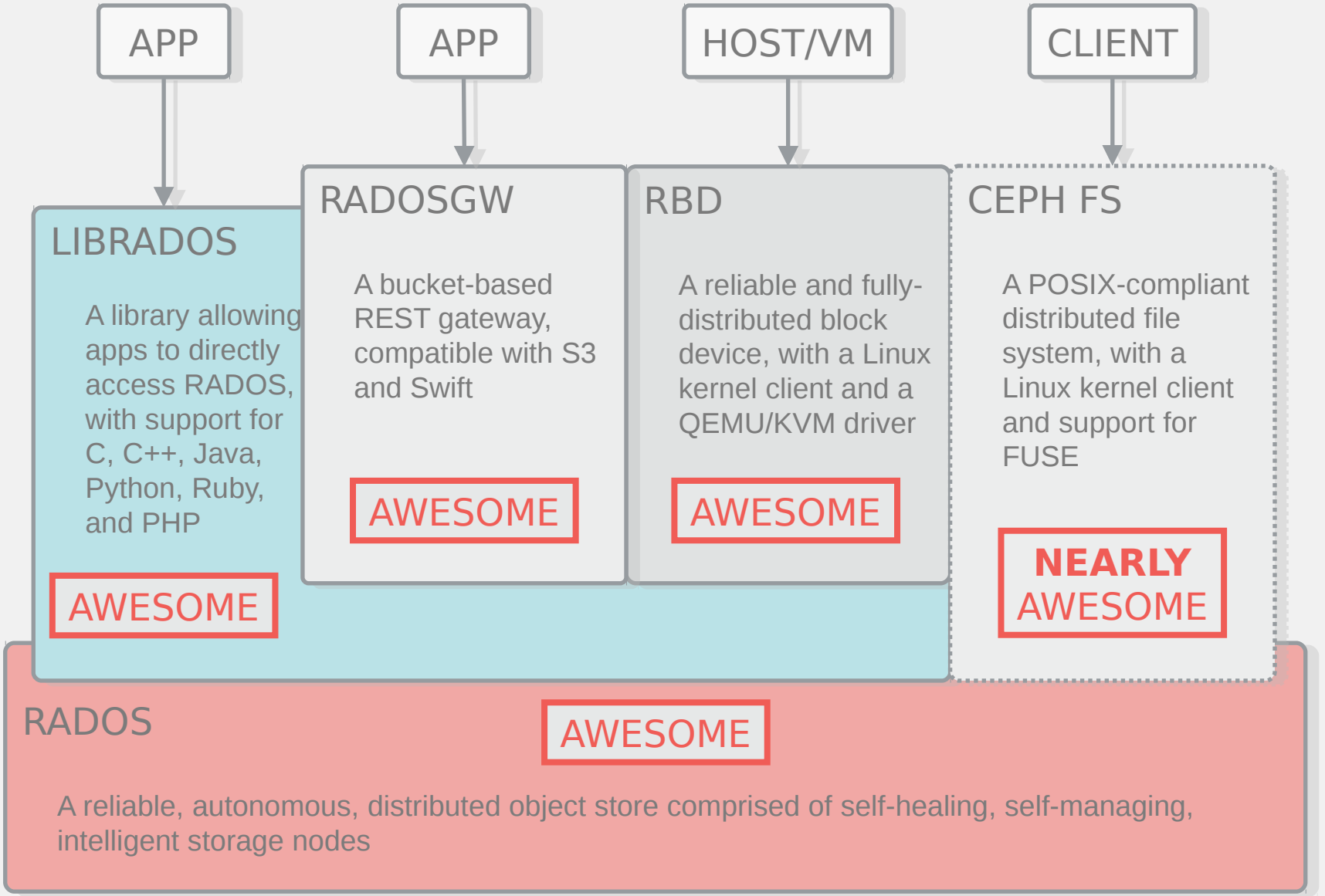
RELEASES

- **Hammer** v0.94.x (LTS)
  - March '15
- Infernalis v9.2.x
  - November '15
- **Jewel** v10.2.x (LTS)
  - April '16
- Kraken v11.2.x
  - November '16
- **Luminous** v12.2.x (LTS)
  - April '17

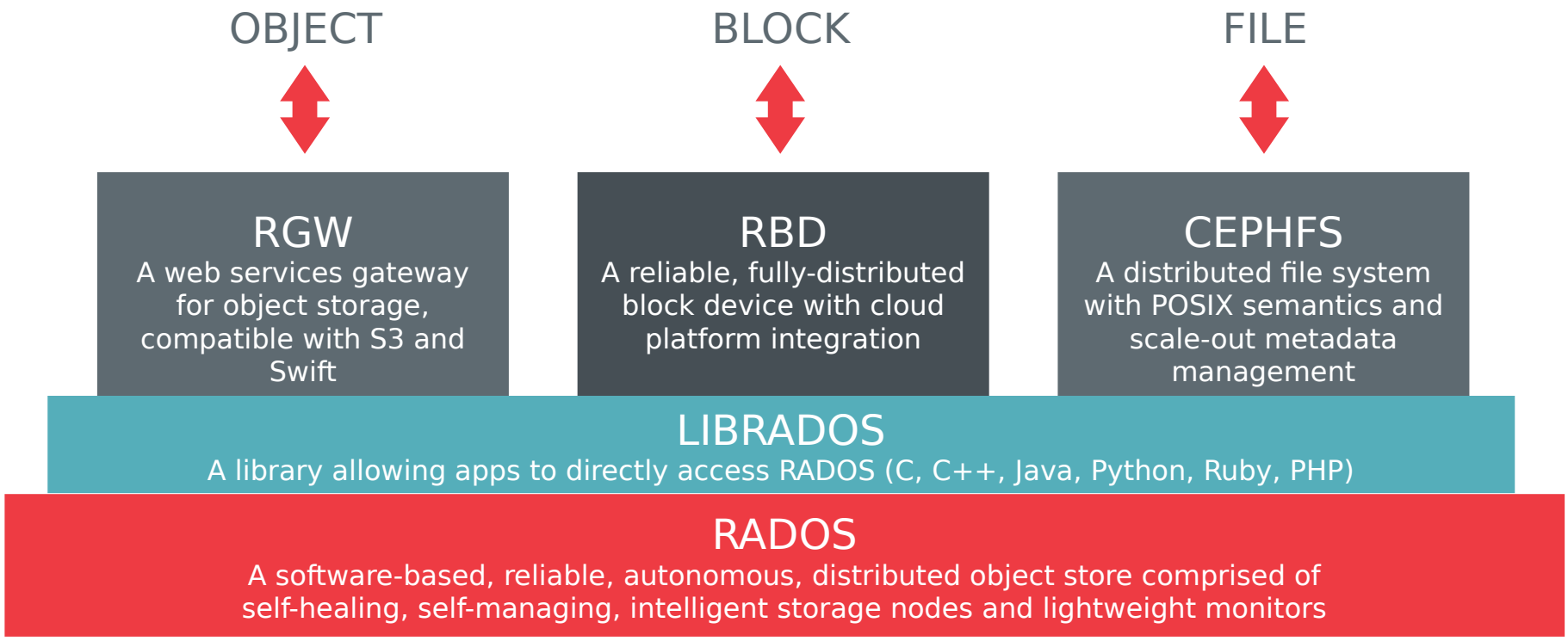
JEWEL v10.2.x – APRIL 2016

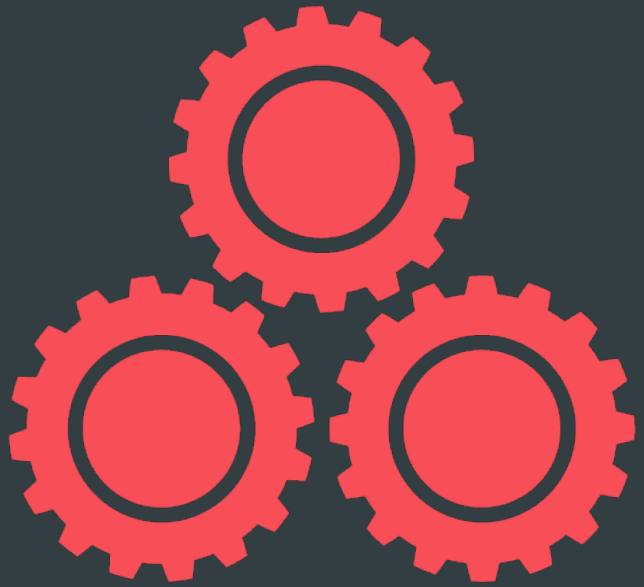


JEWEL


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
AWESOME
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
AWESOME
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
AWESOME
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
NEARLY AWESOME
RADOS
AWESOME
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

2016 = FULLY AWESOME

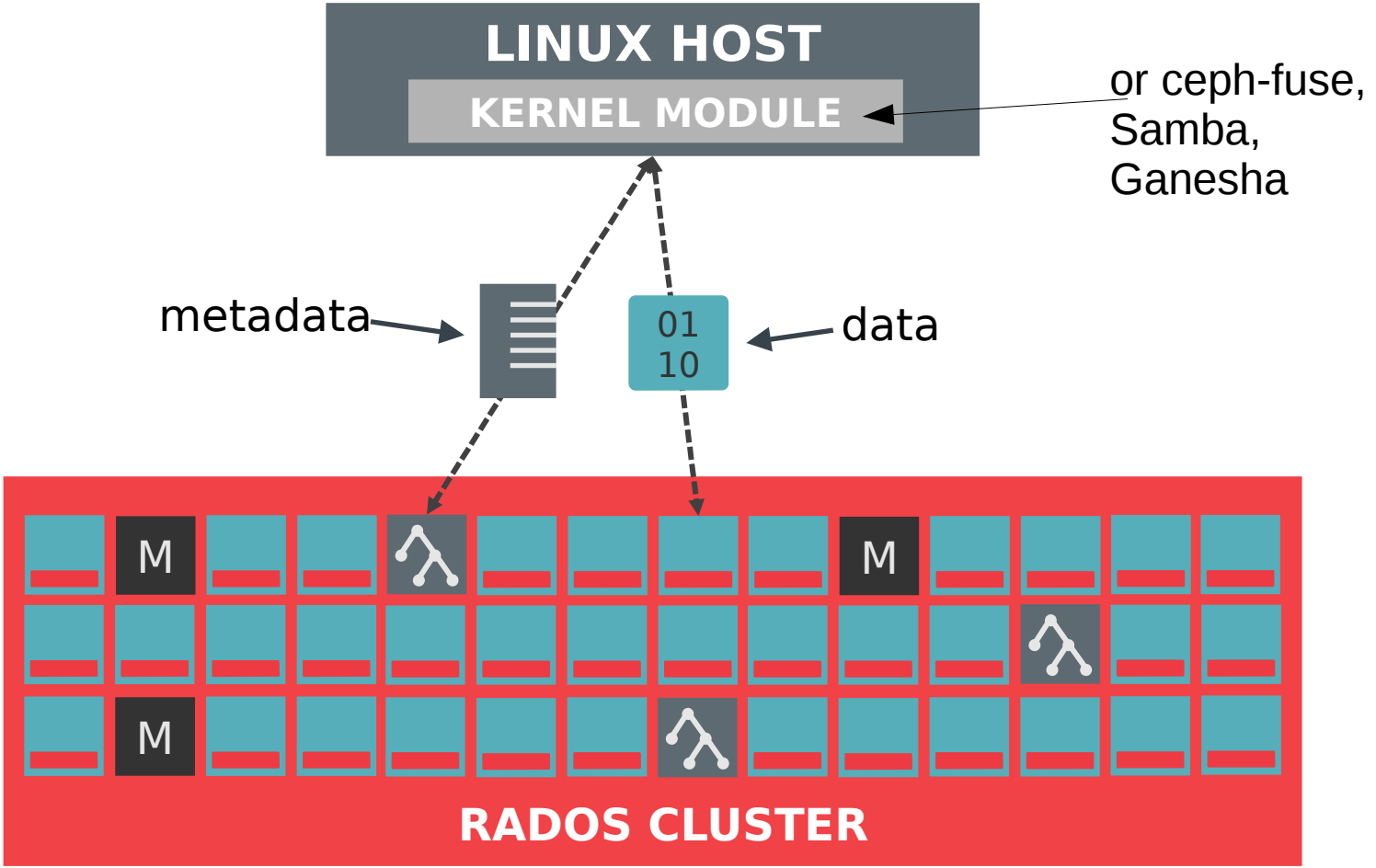




CEPHFS

CEPHFS: STABLE AT LAST

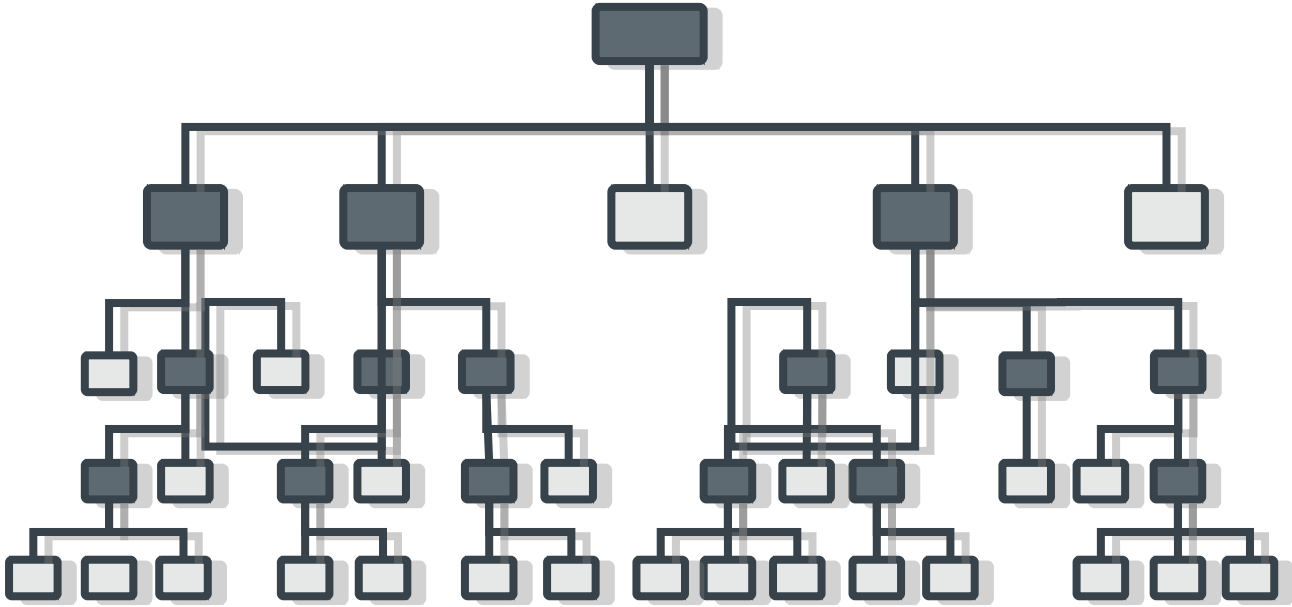
- Jewel recommendations
  - single active MDS (+ many standbys)
  - snapshots disabled
- Repair and disaster recovery tools
- CephFSVolumeManager and Manila driver
- Authorization improvements (confine client to a directory)


LINUX HOST
KERNEL MODULE
or ceph-fuse,
Samba,
Ganesha
metadata
01
10
data
M
M
M
RADOS CLUSTER

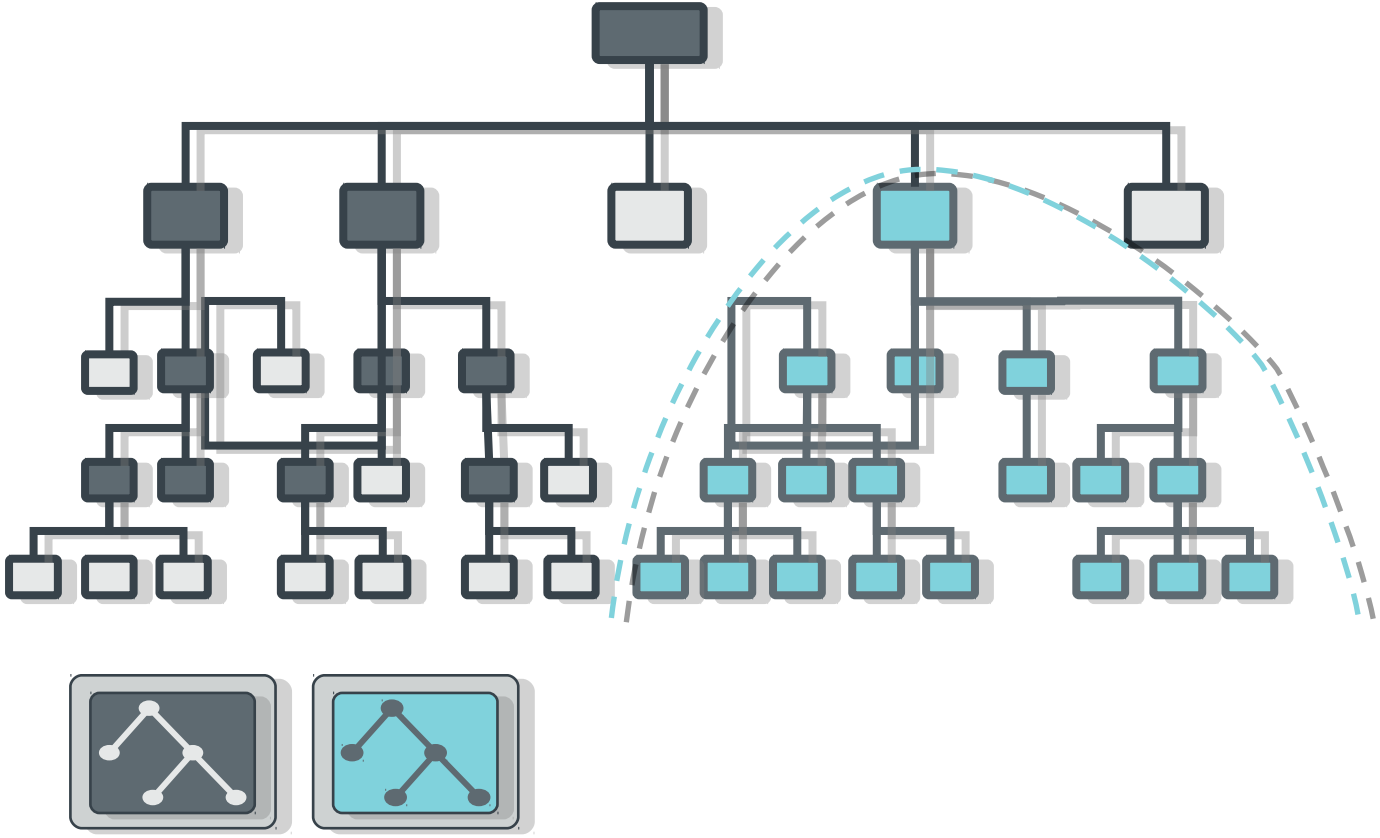
SCALING FILE PEFORMANCE

- Data path is direct to RADOS
  - scale IO path by adding OSDs
  - or use SSDs, etc.
- No restrictions on file count or file system size
  - MDS cache performance related to size of active set, not total file count
- Metadata performance
  - provide lots of RAM for MDS daemons (no local on-disk state needed)
  - use SSDs for RADOS metadata pool
- Metadata path is scaled independently
  - up to 128 active metadata servers tested; 256 possible
  - in Jewel, only 1 is recommended
  - stable multi-active MDS coming in Kraken or Luminous

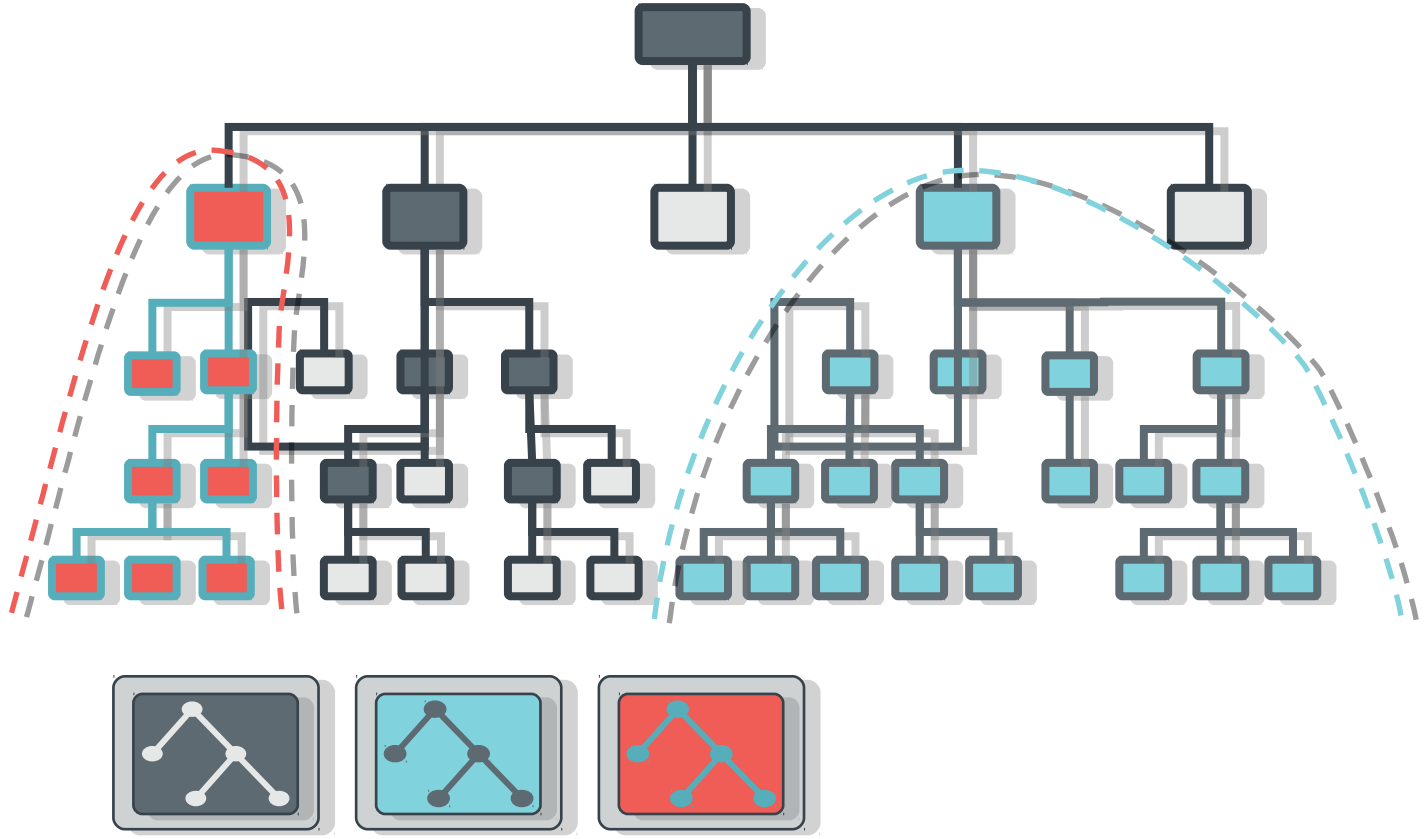
DYNAMIC SUBTREE PARTITIONING



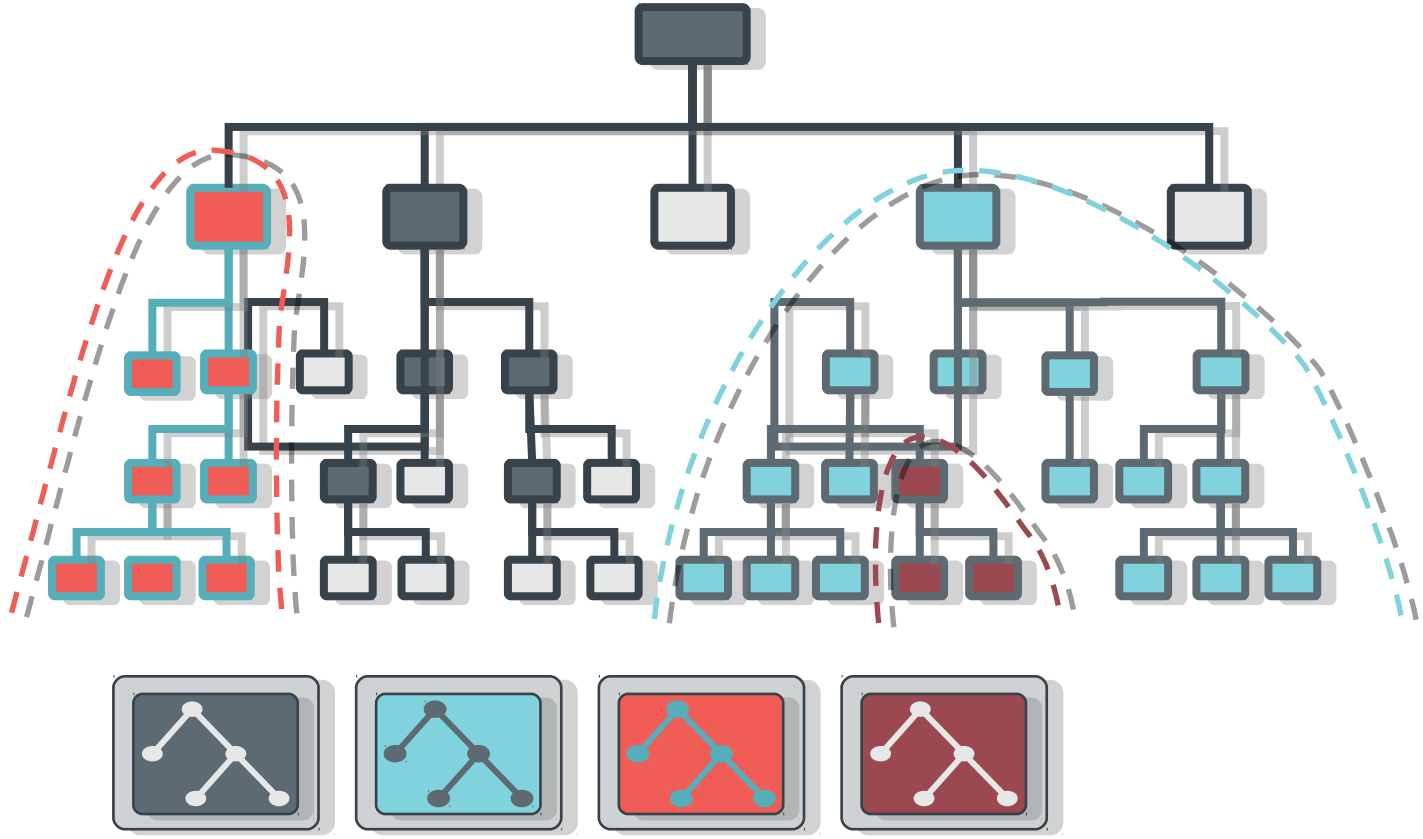
DYNAMIC SUBTREE PARTITIONING



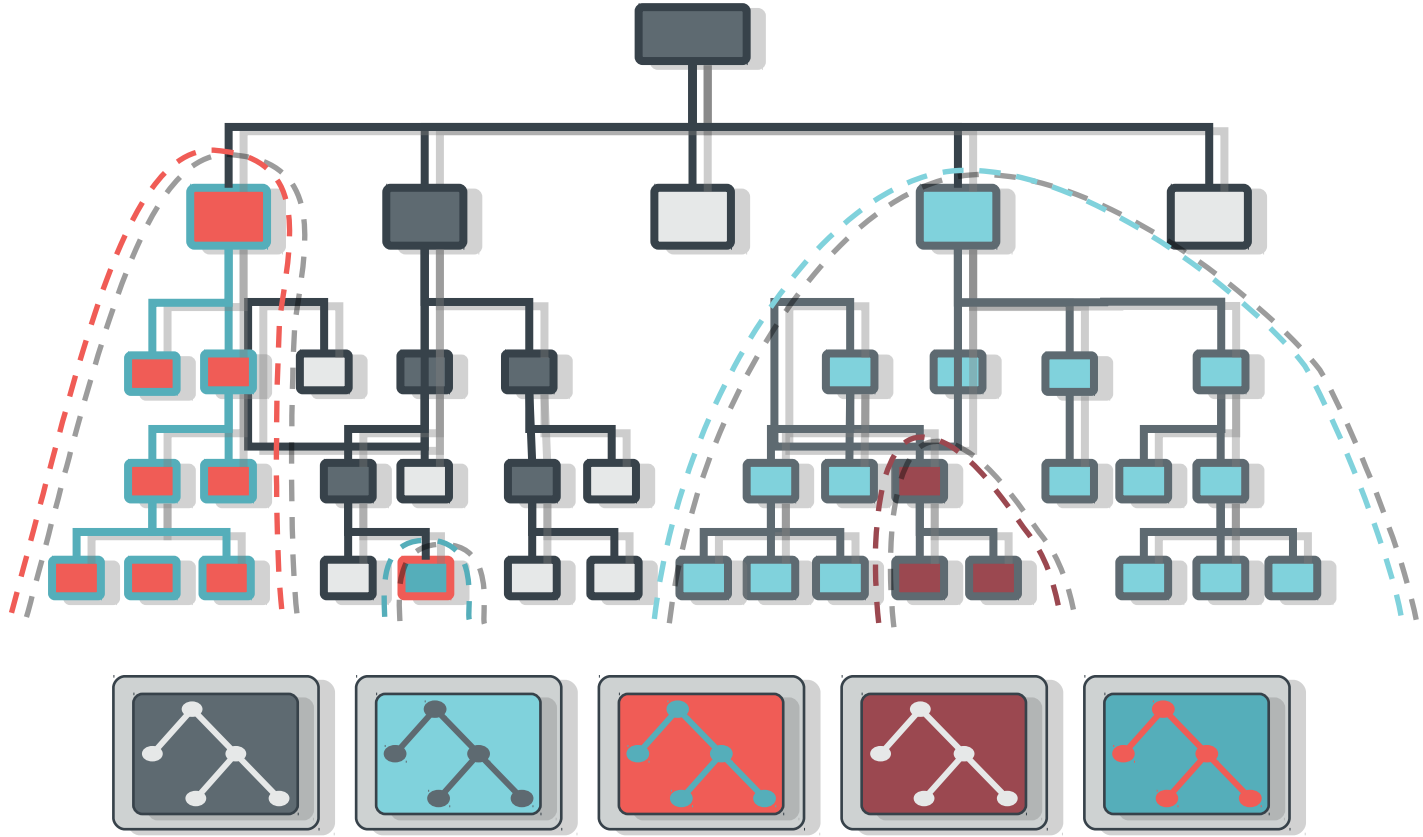
DYNAMIC SUBTREE PARTITIONING



DYNAMIC SUBTREE PARTITIONING



DYNAMIC SUBTREE PARTITIONING



POSIX AND CONSISTENCY

- CephFS has “consistent caching”
  - clients can cache data
  - caches are coherent
  - MDS invalidates data that is changed – complex locking/leasing protocol
- This means clients never see stale data of any kind
  - consistency is much stronger than, say, NFS
- file locks are fully supported
  - flock and fcntl locks

RSTATS

```
# ext4 reports dirs as 4K
ls -lhd /ext4/data
drwxrwxr-x. 2 john john 4.0K Jun 25 14:58 /home/john/data

# cephfs reports dir size from contents
$ ls -lhd /cephfs/mydata
drwxrwxr-x. 1 john john 16M Jun 25 14:57 ./mydata
```

OTHER GOOD STUFF

- Directory fragmentation
  - shard directories for scaling, performance
  - disabled by default in Jewel; on by default in Kraken
- Snapshots
  - create snapshot on any directory

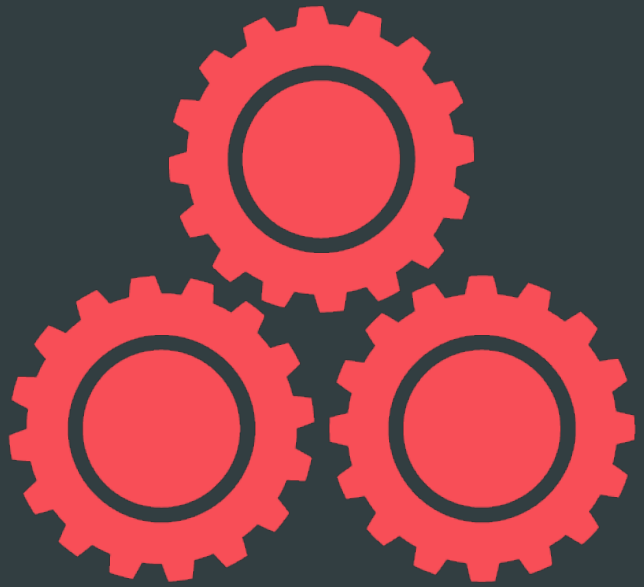
    ```
    mkdir some/random/dir/.snap/mysnapshot
    ls some/random/dir/.snap/mysnapshot
    ```
  - disabled by default in Jewel; hopefully on by default in Luminous
- Security authorization model
  - confine a client mount to a directory and to a rados pool namespace

FSCK AND RECOVERY

- metadata scrubbing
  - online operation
  - manually triggered in Jewel
  - automatic background scrubbing coming in Kraken, Luminous
- disaster recovery tools
  - rebuild file system namespace from scratch if RADOS loses it or something corrupts it

OPENSTACK MANILA FSaaS

- CephFS native
  - Jewel and Mitaka
  - CephFSVolumeManager to orchestrate shares
    - CephFS directories
    - with quota
    - backed by a RADOS pool + namespace
    - and clients locked into the directory
  - VM mounts CephFS directory (ceph-fuse, kernel client, …)
  - tenant VM talks directory to Ceph cluster; deploy with caution



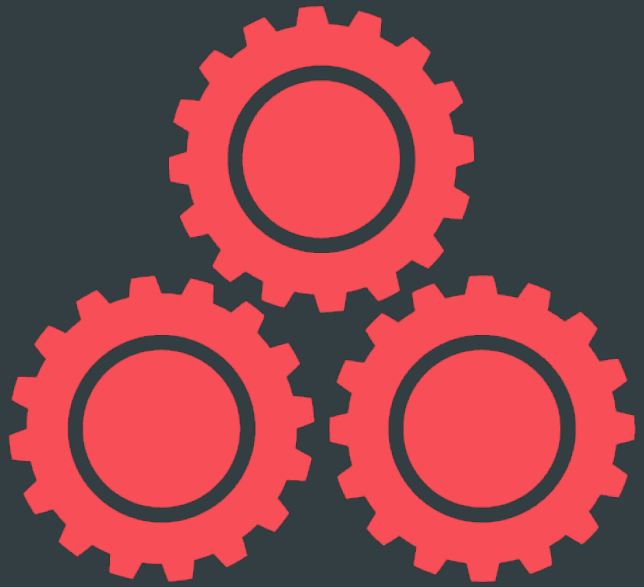
OTHER JEWEL STUFF

GENERAL

- daemons run as ceph user
  - except upgraded clusters that don't want to chown -R
- selinux support
- all systemd
- ceph-ansible deployment
- ceph CLI bash completion
- “calamari on mons”

BUILDS

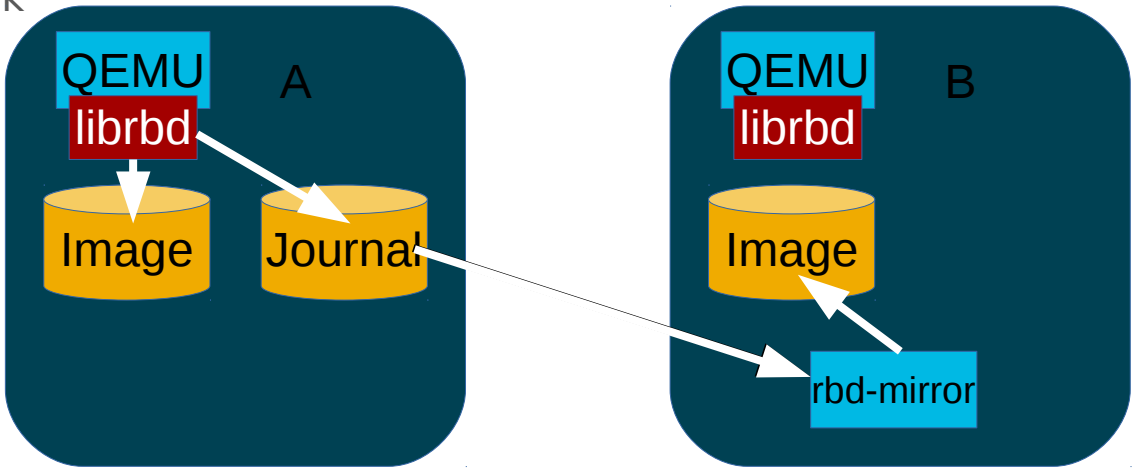
- aarch64 builds
  - centos7, ubuntu xenial
- armv7l builds
  - debian jessie
  - http://ceph.com/community/500-osd-ceph-cluster/



RBD

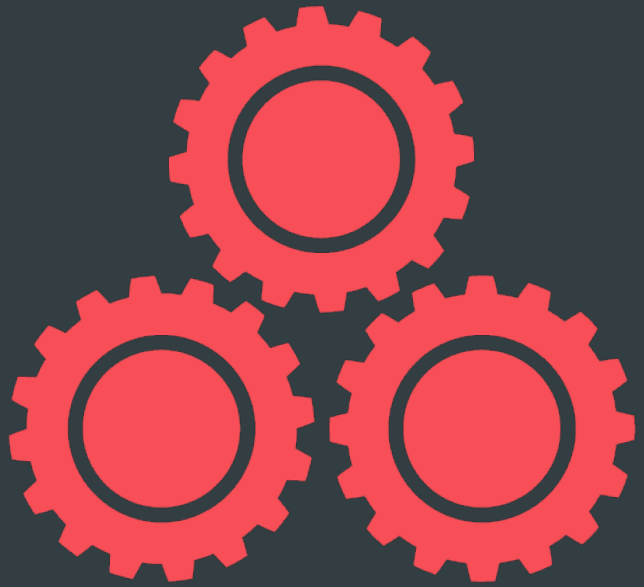
RBD IMAGE MIRRORING

- image mirroring
  - asynchronous replication to another cluster
  - replica(s) crash consistent
  - replication is per-image
  - each image has a data journal
  - rbd-mirror daemon does the work



OTHER RBD STUFF

- fast diff
  - use object-map to do O(1) time diff
- deep flatten
  - separate clone from parent while retaining snapshot history
- dynamic features
  - turn on/off: exclusive-lock, object-map, fast-diff, journaling
  - turn off: deep-flatten
  - useful for compatibility with kernel client, which lacks some new features
- new default features
  - layering, exclusive-lock, object-map, fast-diff, deep-flatten
- snapshot rename
- rbd du
- improved/rewritten CLI (with dynamic usage/help)



RGW

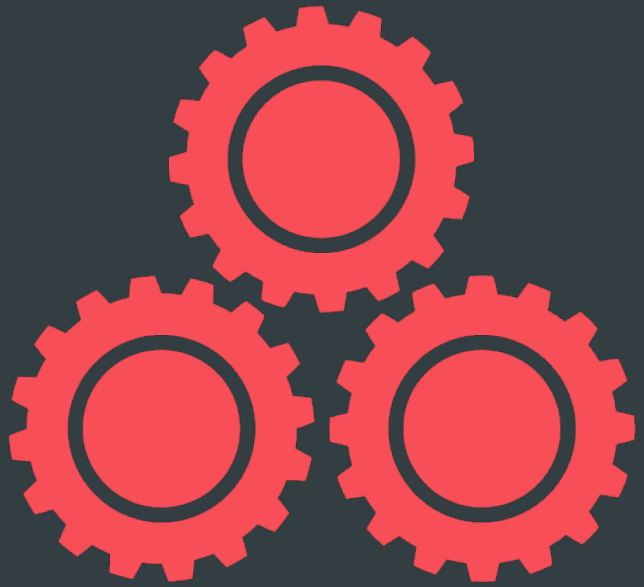
NEW IN RGW

- Rewritten multi-site capability
  - N zones, N-way sync
  - fail-over and fail-back
  - simpler configuration
- NFS interface
  - export a bucket over NFSv4
  - designed for import/export of data – not general a purpose file system!
  - based on nfs-ganesha
- Indexless buckets
  - bypass RGW index for certain buckets

    (that don't need enumeration, quota, ...)

RGW API UPDATES

- S3
  - AWS4 authentication support
  - LDAP and AD/LDAP support
  - static website
  - RGW STS (coming shortly)
    - Kerberos, AD integration
- Swift
  - Keystone V3
  - multi-tenancy
  - object expiration
  - Static Large Object (SLO)
  - bulk delete
  - object versioning
  - refcore compliance



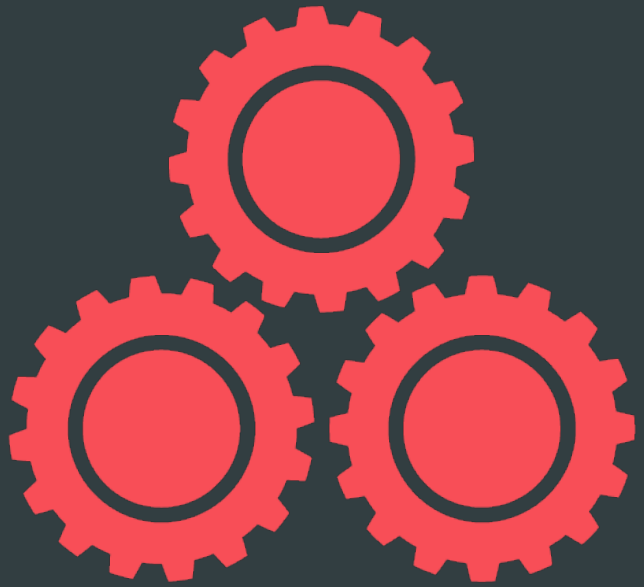
RADOS

RADOS

- queuing improvements
  - new IO scheduler “wpq” (weighted priority queue) stabilizing
  - (more) unified queue (client io, scrub, snaptrim, most of recovery)
  - somewhat better client vs recovery/rebalance isolation
- mon scalability and performance improvements
  - thanks to testing here @ CERN
- optimizations, performance improvements (faster on SSDs)
- AsyncMessenger – new implementation of networking layer
  - fewer threads, friendlier to allocator (especially tcmalloc)

MORE RADOS

- no more ext4
- cache tiering improvements
  - proxy write support
  - promotion throttling
  - better, still not good enough for RBD and EC base
- SHEC erasure code (thanks to Fujitsu)
  - trade some extra storage for recovery performance
- [test-]reweight-by-utilization improvements
  - more better data distribution optimization
- BlueStore – new experimental backend



BLUESTORE

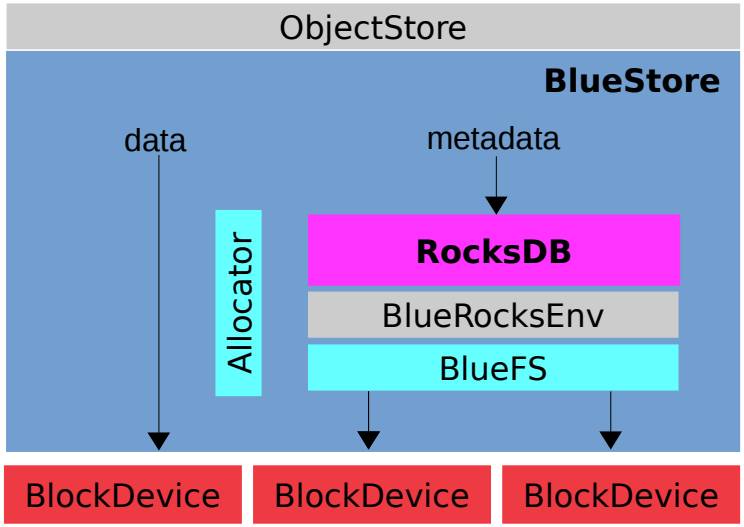
BLUESTORE: NEW BACKEND GOALS

- no more double writes to a full data journal
- efficient object enumeration
- efficient clone operation
- efficient splice (“move these bytes from object X to object Y”)
  - (will enable EC overwrites for RBD over EC)
- efficient IO pattern for HDDs, SSDs, NVMe
- minimal locking, maximum parallelism (between PGs)
- full data and metadata checksums
- inline compression (zlib, snappy, etc.)

BLUESTORE

- BlueStore = **Bl**ock + N**ewStore**
  - consume raw block device(s)
  - key/value database (RocksDB) for metadata
  - data written directly to block device
  - pluggable block Allocator

- We must share the block device with RocksDB
  - implement our own rocksdb::Env
  - implement tiny “file system” BlueFS
  - make BlueStore and BlueFS share



WE WANT FANCY STUFF

Full data checksums

- We scrub... periodically
- We want to validate checksum on **every** read

Compression

- 3x replication is expensive
- Any scale-out cluster is expensive

WE WANT FANCY STUFF

Full data checksums

- We scrub... periodically
- We want to validate checksum on **every** read

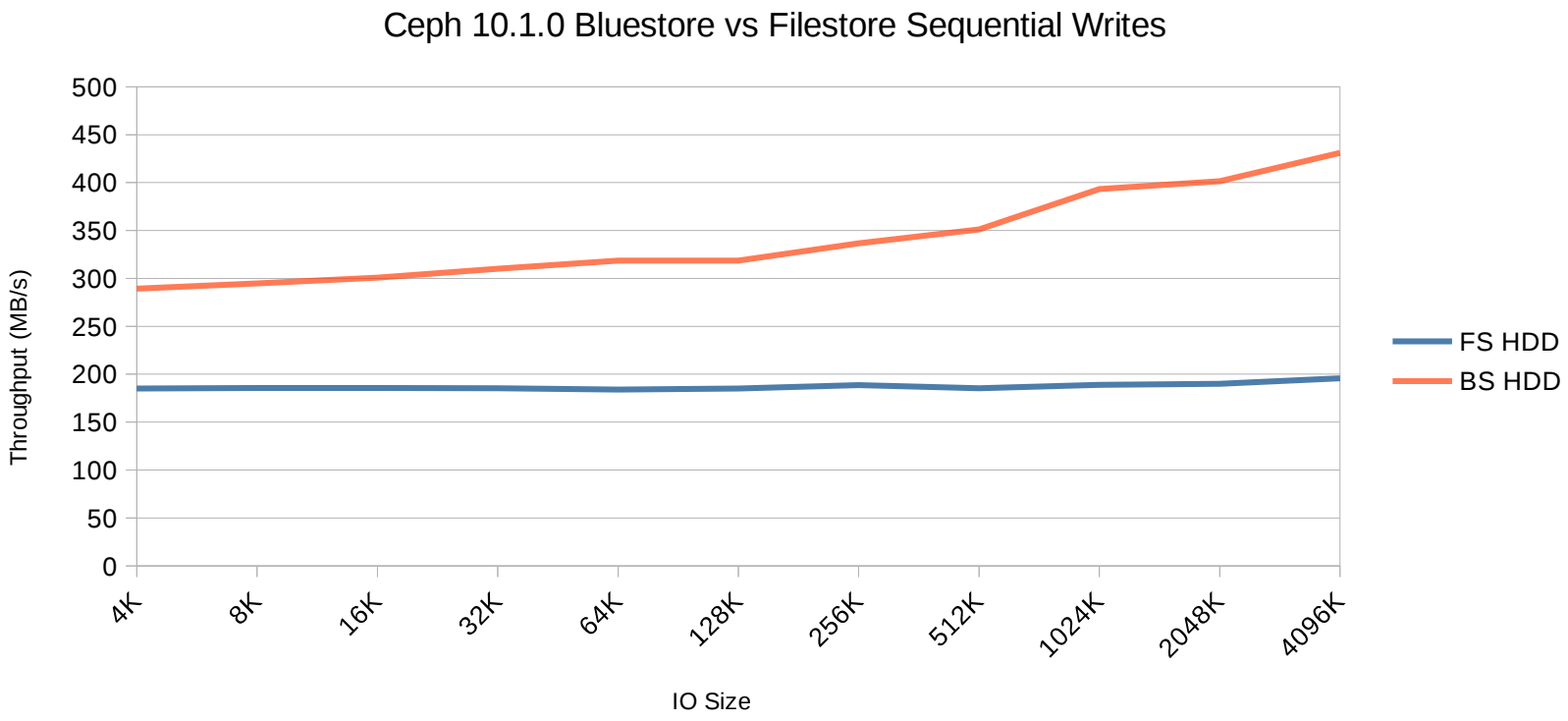
- More metadata in the blobs
  - 4KB of 32-bit csum metadata for 4MB object and 4KB blocks
  - larger csum blocks?
  - smaller csums (8 or 16 bits)?

Compression

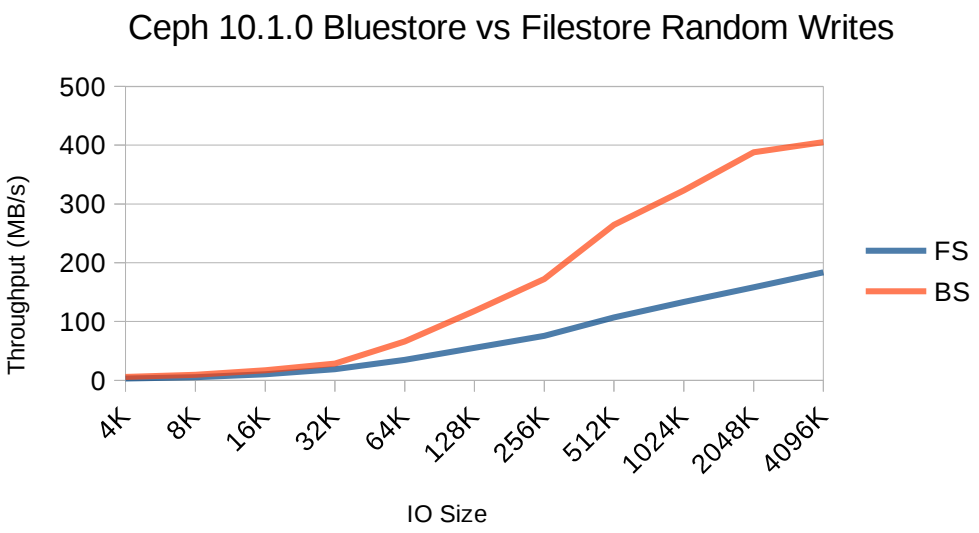
- 3x replication is expensive
- Any scale-out cluster is expensive

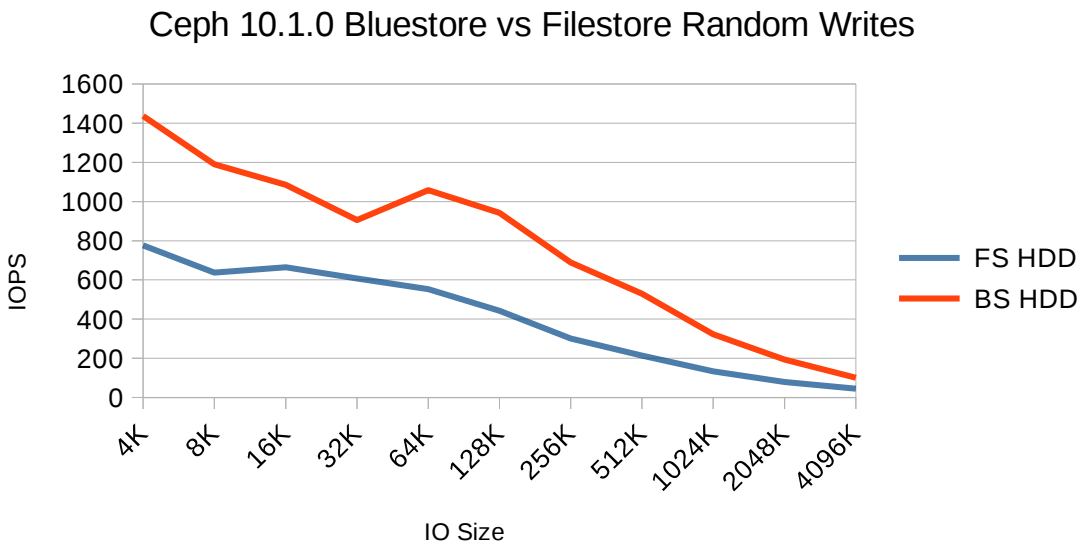
- Need largish extents to get compression benefit (64 KB, 128 KB)
  - overwrites occlude/obscure compressed blobs
  - compacted (rewritten) when >N layers deep

HDD: SEQUENTIAL WRITE

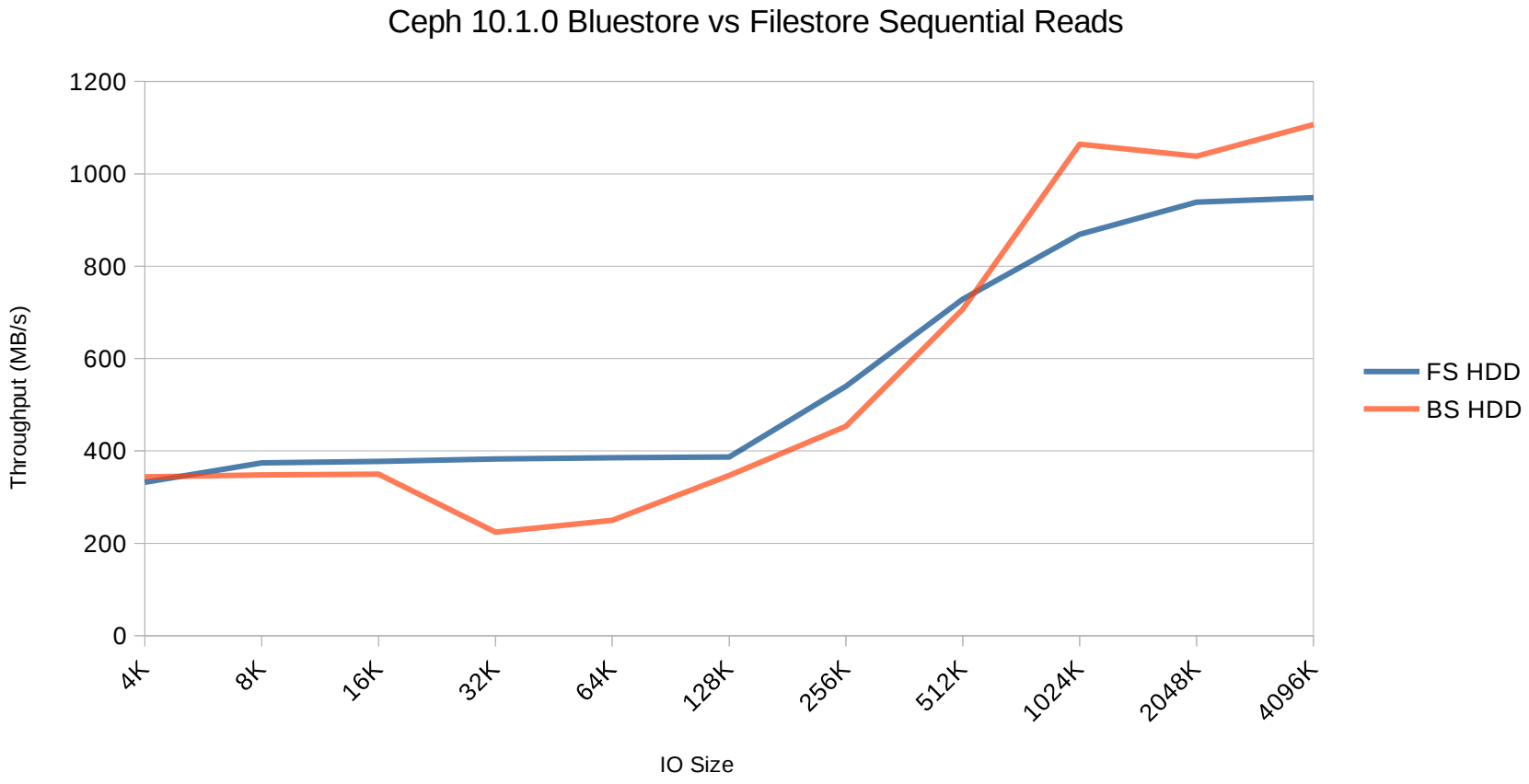


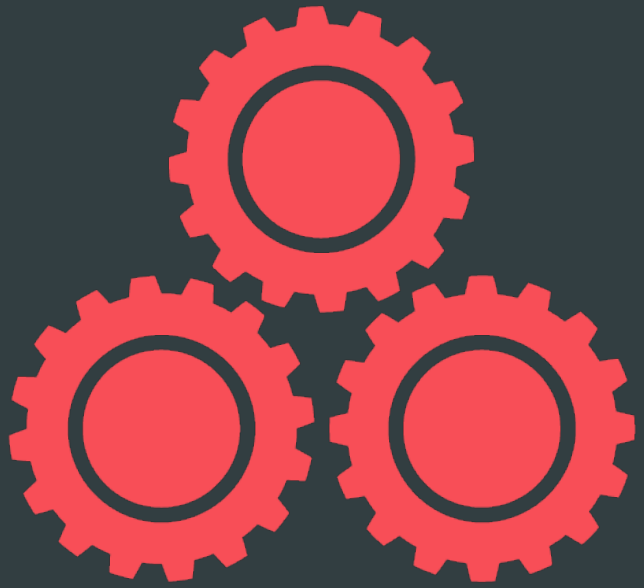
HDD: RANDOM WRITE





HDD: SEQUENTIAL READ





KRAKEN AND LUMINOUS

RADOS

- BlueStore!
- erasure code overwrites (RBD + EC)
- ceph-mgr – new mon-like daemon
  - management API endpoint (Calamari)
  - metrics
- config management in mons
- on-the-wire encryption
- OSD IO path optimization
- faster peering
- QoS
- ceph-disk support for dm-cache/bcache/FlashCache/...

RGW

- AWS STS (kerberos support)
- pluggable full-zone syncing
  - tiering to tape
  - tiering to cloud
  - metadata indexing (elasticsearch?)
- S3 encryption API
- compression
- performance

RBD

- RBD mirroring improvements
  - cooperative daemons
  - Cinder integration
- RBD client-side persistent cache
  - write-through and write-back cache
  - ordered writeback → crash consistent on loss of cache
- RBD consistency groups
- client-side encryption
- Kernel RBD improvements

CEPHFS

- multi-active MDS

  and/or

- snapshots
- Manila hypervisor-mediated FsaaS
  - NFS over VSOCK →

    libvirt-managed Ganesha server →

    libcephfs FSAL →

    CephFS cluster
  - new Manila driver
  - new Nova API to attach shares to Vms
- Samba and Ganesha integration improvements
- richacl (ACL coherency between NFS and CIFS)

OTHER COOL STUFF

- librados backend for RocksDB
  - and rocksdb is not a backend for MySQL...
- PMStore
  - Intel OSD backend for 3D-Xpoint
- multi-hosting on IPv4 and IPv6
- ceph-ansible
- ceph-docker

GROWING DEVELOPMENT COMMUNITY

- SUSE
- Mirantis
- XSKY
- Intel
- Fujitsu
- DreamHost
- ZTE
- SanDisk
- Gentoo
- Samsung
- LETV
- Igalia
- Deutsche Telekom
- Kylin Cloud
- Endurance International Group
- H3C
- Johannes Gutenberg-Universität Mainz
- Reliance Jio Infocomm
- Ebay
- Tencent

THANK YOU!

Sage Weil
CEPH PRINCIPAL ARCHITECT


sage@redhat.com


@liewegas