Norfolk, Virginia, USA • **May 8-13, 2023**

CHEP 2023

Computing in High Energy & Nuclear Physics

Enabling Storage Business Continuity and Disaster Recovery with Ceph distributed storage

Enrico Bocchi
Abhishek Lekshmanan
Roberto Valverde

May 8, 2023

Ceph at CERN

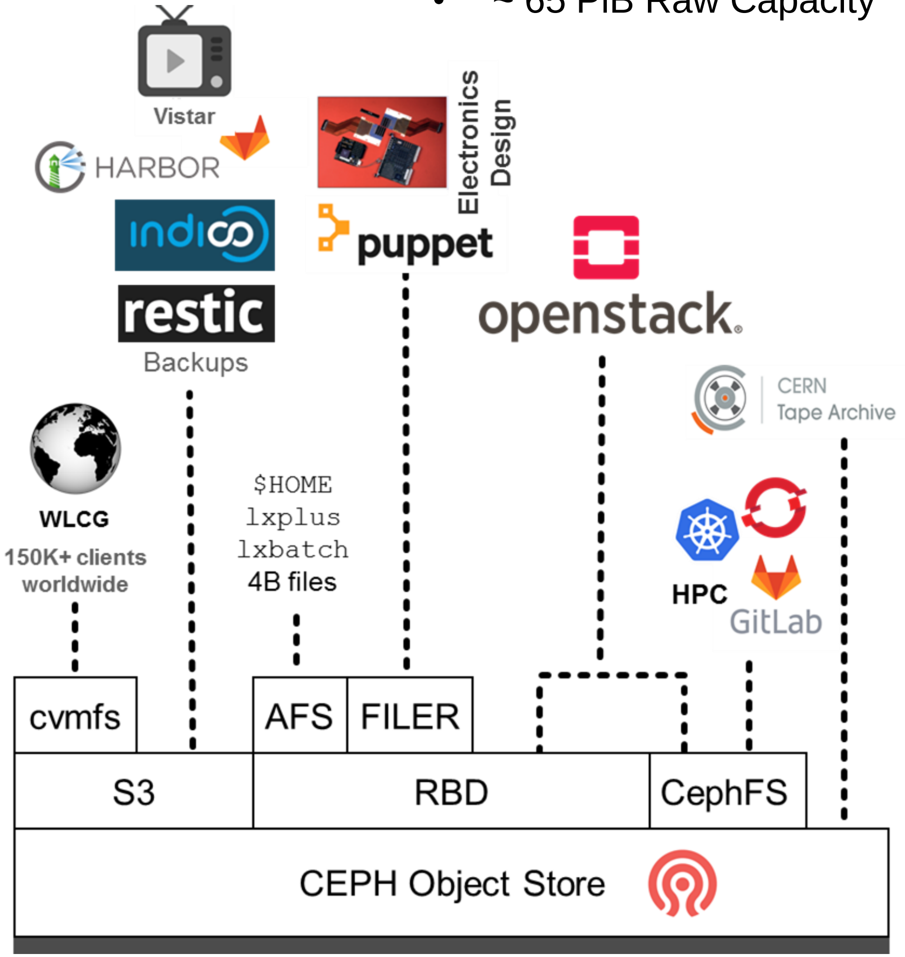
- **Ceph provides 3 types of storage**
  - **Block** – RBD, OpenStack Cinder/Glance Volumes
  - **Object** – S3, Swift
  - **File System** – CephFS, OpenStack Manila Shares, K8s/OKD, HPC scratch

- **IT Services**
  - Cloud Infrastructure, Code repositories, Container Registries, Agile Infra
  - Monitoring: Open Search, Kafka, Gafana, InfluxDB, Kibana
  - Document Repositories // Web: Indico, Drupal, WordPress
  - Analytics: HTCondor, Slurm, Jupyter Notebooks, Apache Spark

- **Other Storage**
  - NFS Filers, AFS, CVMFS, CERN Tape Archive, …

- **Physics Experiments and End-Users**
  - ATLAS Event Index, Alice O2 Build/CI, Microelectronics Design, …

- 16 Production Clusters
- ~ 65 PiB Raw Capacity



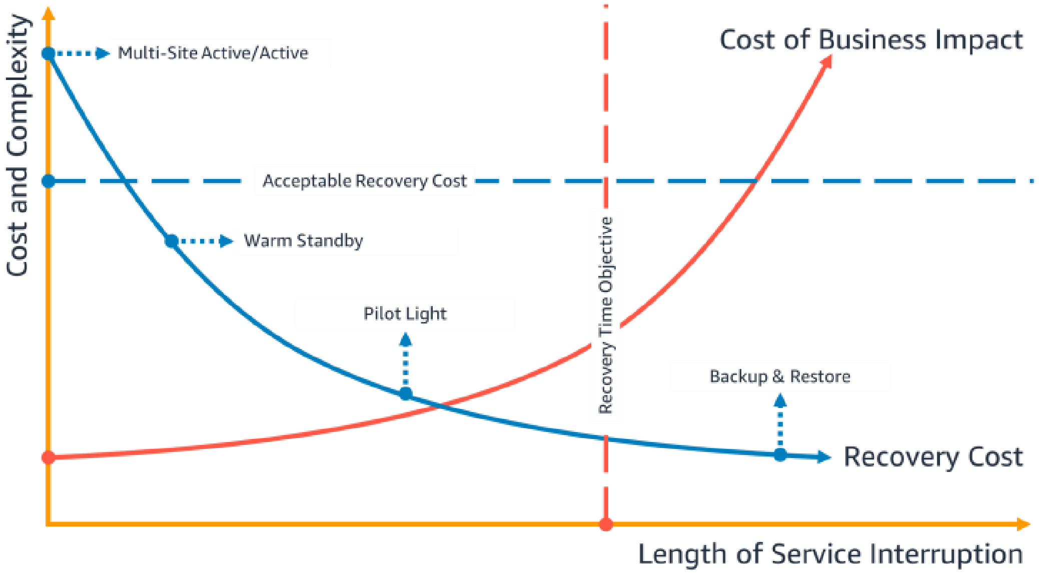
Planning for Ceph BC/DR

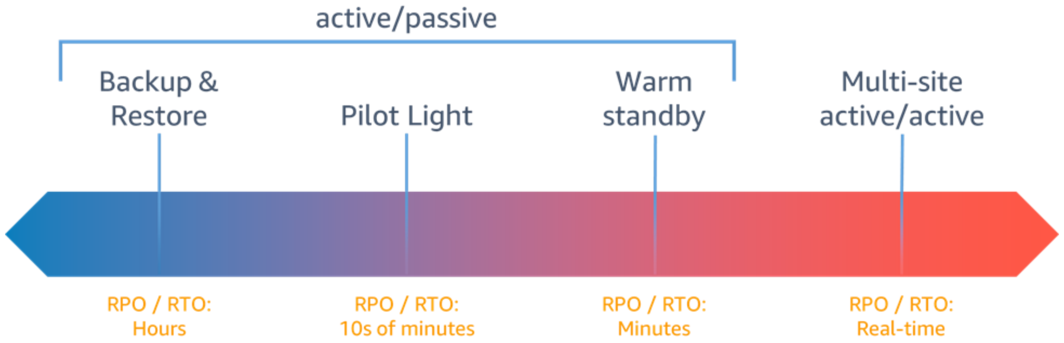
- **Various strategies possible**
  - Active/Active, Active/Passive, Backup & Restore

  - Ceph has features mapping to each strategy
  - Complexity comes from combinations of strategies and storage types (block, object, fs)

- **Driving factors**
  - Use existing components and expertise (upstream and in-house)
  - Technology maturity and reliability
    - Not all Ceph features are immediately production-ready

docs.aws.amazon.com





Purpose of This Talk

This is a journey through our explorations for
Ceph Business Continuity and Disaster Recovery (BC/DR)

- We report on the experience collected while testing Ceph features
- Goal is to collect evidence for decision-making,
  then promote to production the most appropriate solutions according to the requirements

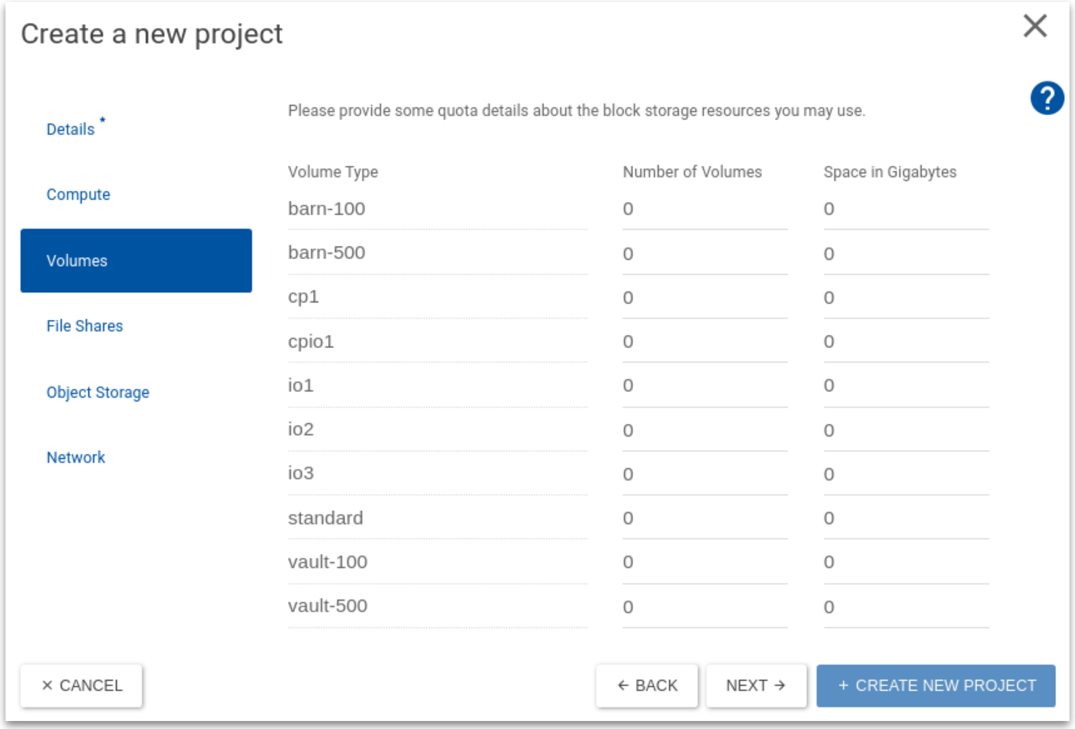
1

RBD, Block Volumes

1. RBD: Storage Availability Zones

- **What for:** BC – High(er) Availability

- **Spread RBDs over multiple clusters**
  - Following major outage, causing 8hrs downtime
    - Evolved from 1 RBD cluster, 4 volume types, to 5 RBD clusters
    - Each cluster is fully decoupled from the others
  - Admittedly less practical to manage and use
    - We (almost) exposed 12 volume types
    - …and a form with 30 fields to fill



Create a new project

Please provide some quota details about the block storage resources you may use.

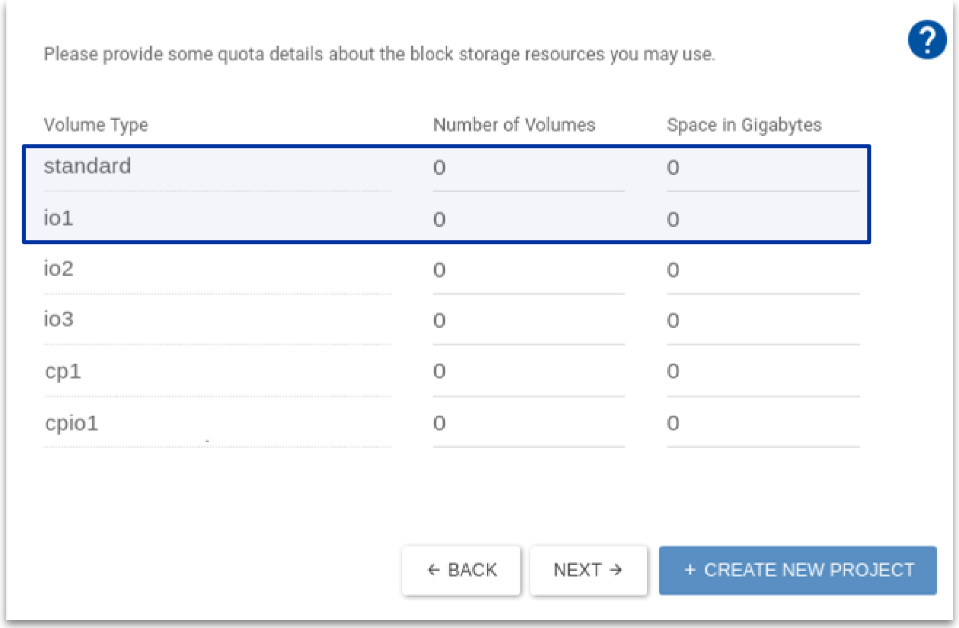
Volume Type	Number of Volumes	Space in Gigabytes
barn-100	0	0
barn-500	0	0
cp1	0	0
cpio1	0	0
io1	0	0
io2	0	0
io3	0	0
standard	0	0
vault-100	0	0
vault-500	0	0

1. RBD: Storage Availability Zones

- **Consolidate volume types according to QoS**
  - Simplify to 6 types exposed to users
  - **Storage Availability Zones** for `standard` and `io1` types
    - Backed by 3 RBD clusters
    - Different rooms, UPSs, network branches

- **Users to decide which Storage AZ hosts the volume**
  - Else, OpenStack Cinder picks a cluster according to internal weighting functions (e.g., least full cluster)

```
$ openstack volume create --size 10 \
             --availability-zone ceph-geneva-3 chep23
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| availability_zone   | ceph-geneva-3                        |
| name                | chep23                         |
| size                | 10                                   |
| status              | creating                             |
| type                | standard                             |
+---------------------+--------------------------------------+
```



Please provide some quota details about the block storage resources you may use.

Volume Type	Number of Volumes	Space in Gigabytes
standard	0	0
io1	0	0
io2	0	0
io3	0	0
cp1	0	0
cpio1	0	0

1. RBD: Backups

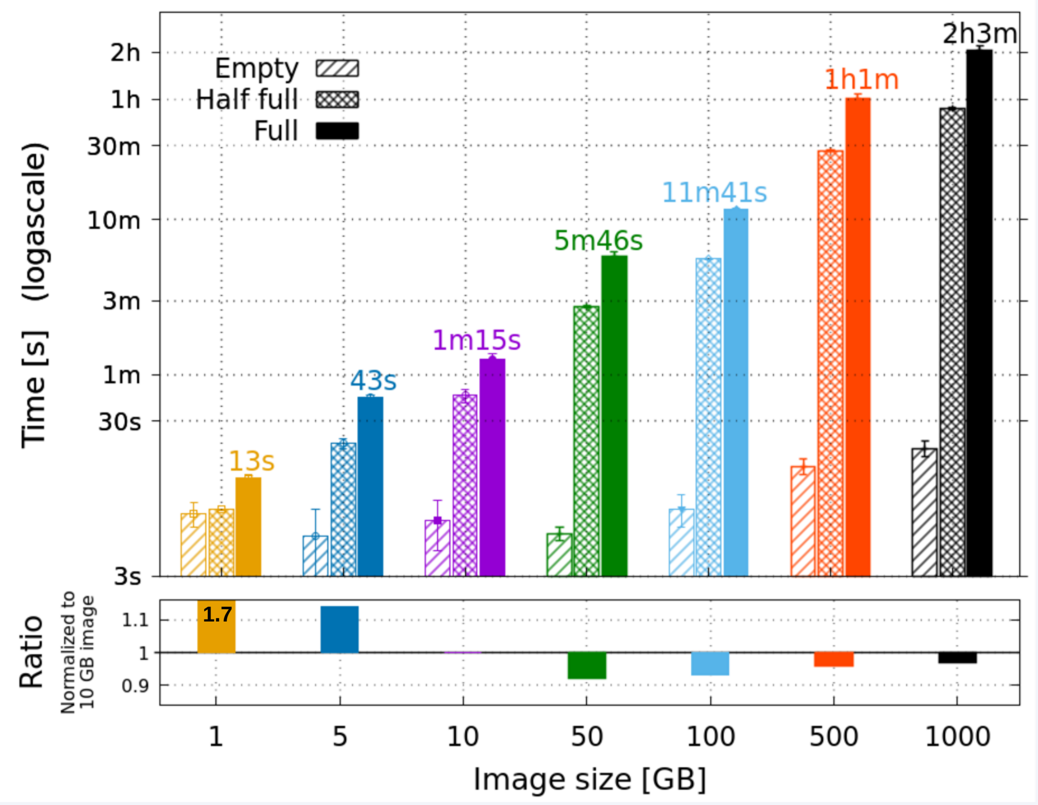
- **What for:** DR – Backup & Restore

- **Full Backups, rbd-to-rbd**
  - Relies on `librbd` and low-level RBD features

```
`rbd export-diff | rbd import-diff`
```

  - Good backup performance out-of-the-box:
    - RBD copies at ~140 MB/s per image
    - Speed is sustained and consistent with varying image sizes

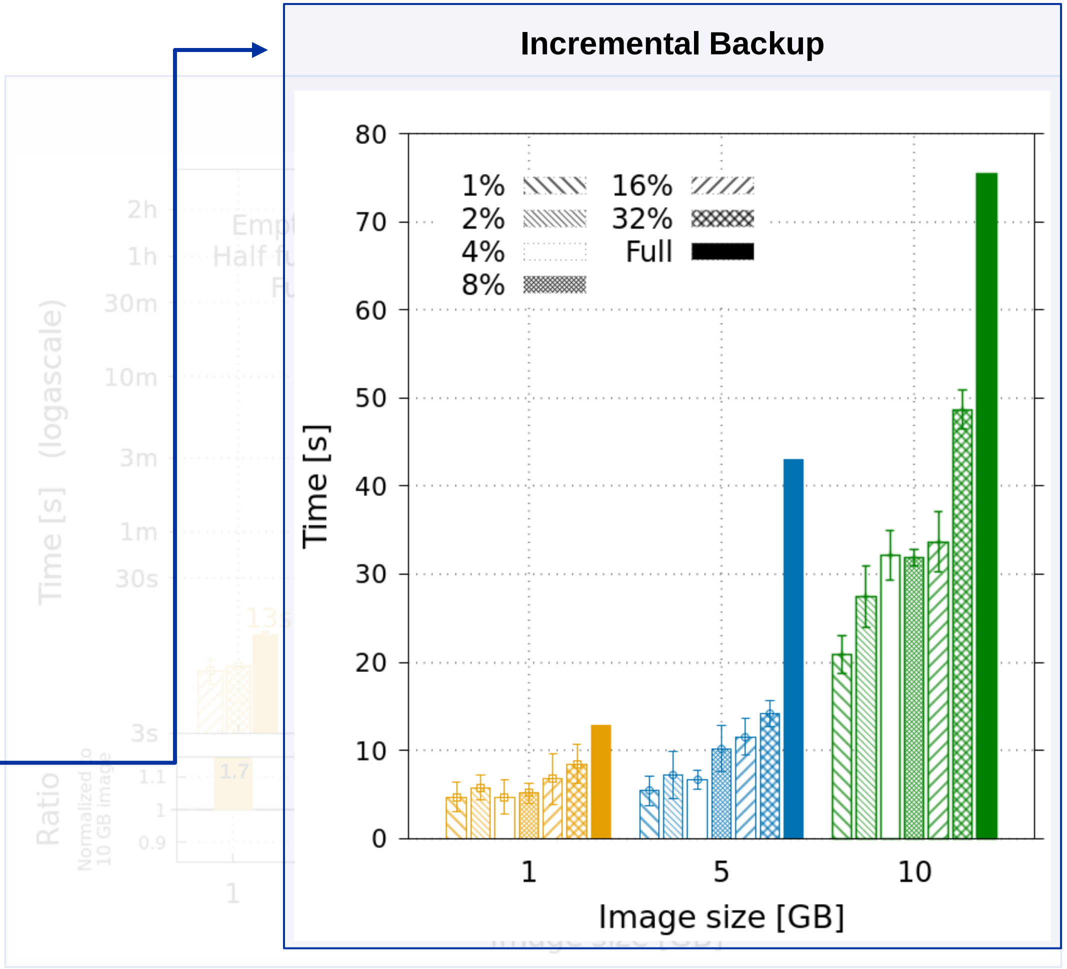
Full Backup



1. RBD: Backups

- **What for:** DR – Backup & Restore

- **Full Backups, rbd-to-rbd**
  - Relies on `librbd` and low-level RBD features
    `` `rbd export-diff | rbd import-diff` ``
  - Good backup performance out-of-the-box:
    - RBD copies at ~140 MB/s per image
    - Speed is sustained and consistent with varying image sizes
  - Efficient incremental backups:
    - Based on difference (`fast-diff`, `object-map`) between previous backup and current state of the image
    - Copy only the extents that changed to backup target



2

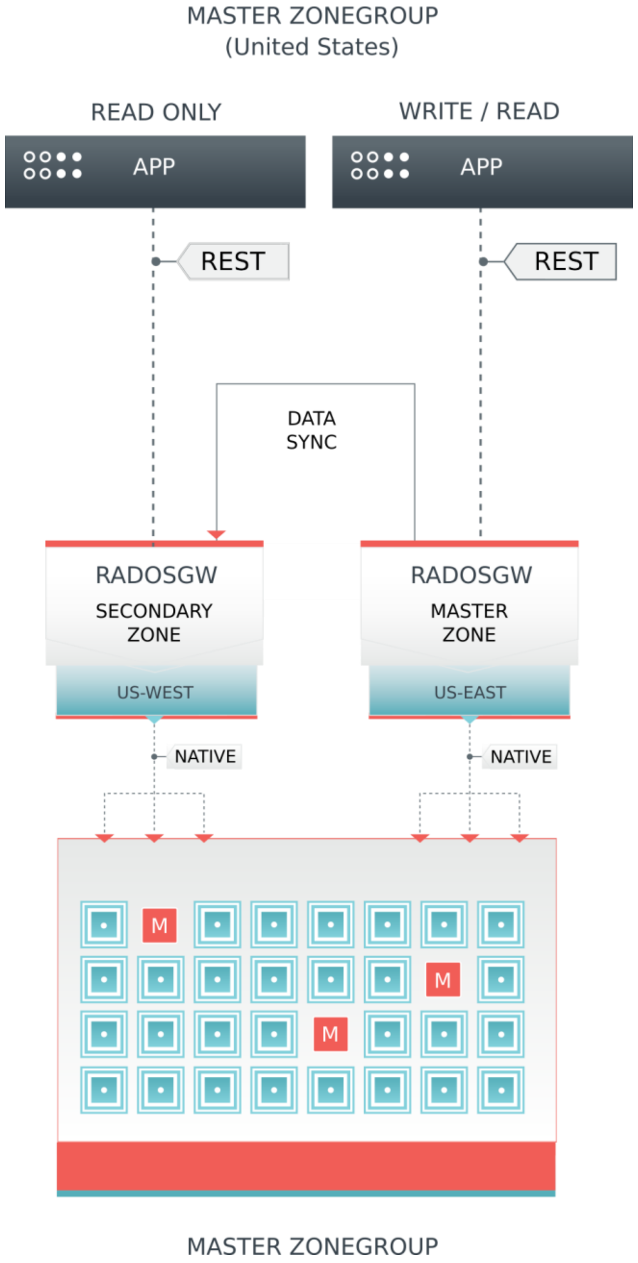
S3 Objects

2. S3 Objects: Multisite Replication

- **What for:** BC – High(er) Availability

- **Full mirror with master + secondary zone**
  - Test setup with 2 bare-metal clusters (Quincy 17.2.5)
  - Two zones (`rw`), one zonegroup,
    dedicated `radosgws` for sync traffic configured as zone endpoints

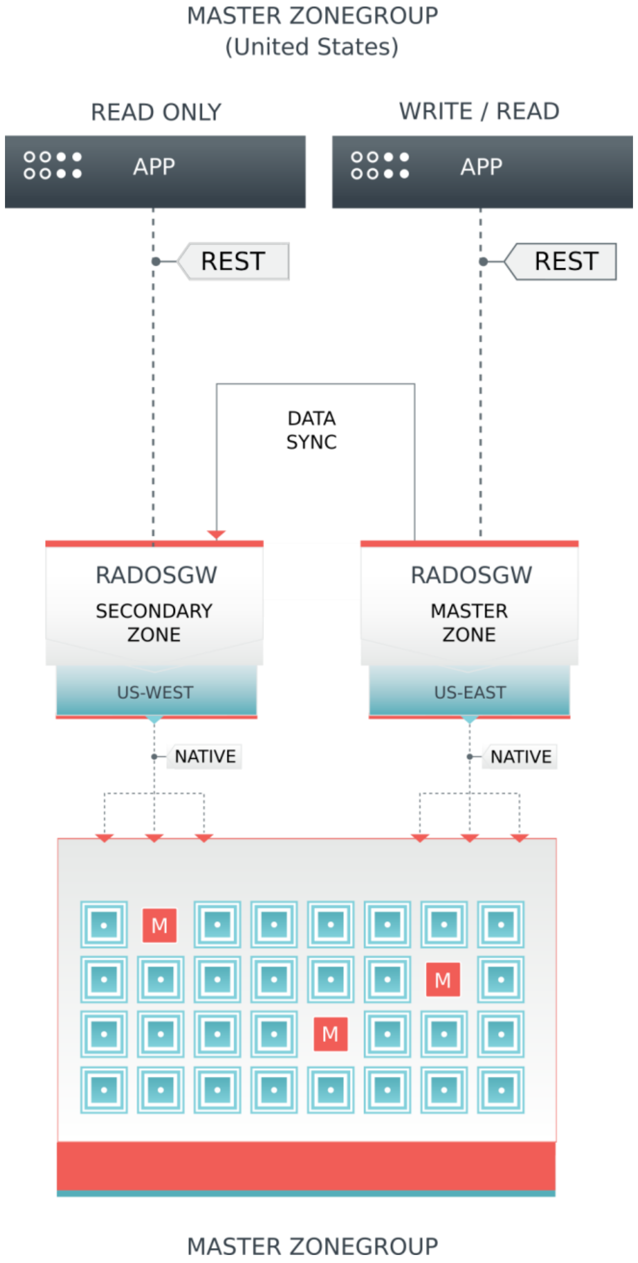
- **Basic functional testing with MinIO Warp**
  - 1M objects, log2 random size (up to 64 MB), multipart uploads
    - Very flexible: Distribution of request types, versioning, retention, ranges, …
  - Not specific to multisite deployments



2. S3 Objects: Multisite Replication

- **Main pain points**

  1. Sync may lag behind and struggles to recover
     - We wrote 1M objects to the master zone, while secondary was shut-off
     - It took ~1 day to sync with no other load on the clusters



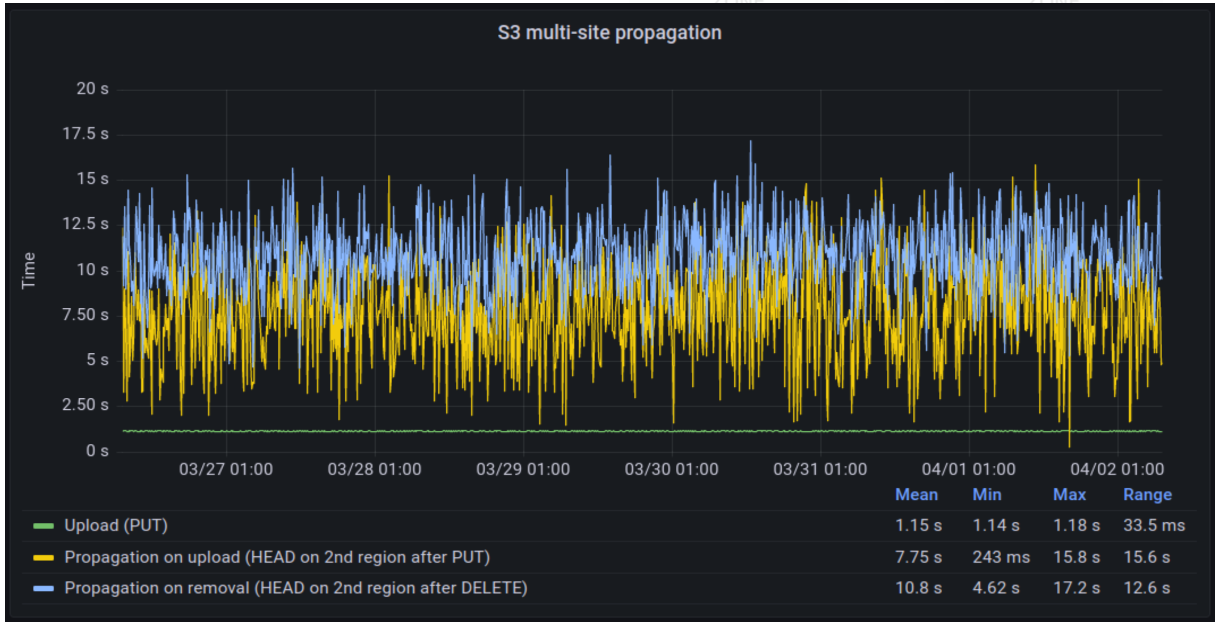
2. S3 Objects: Multisite Replication

- **Main pain points**

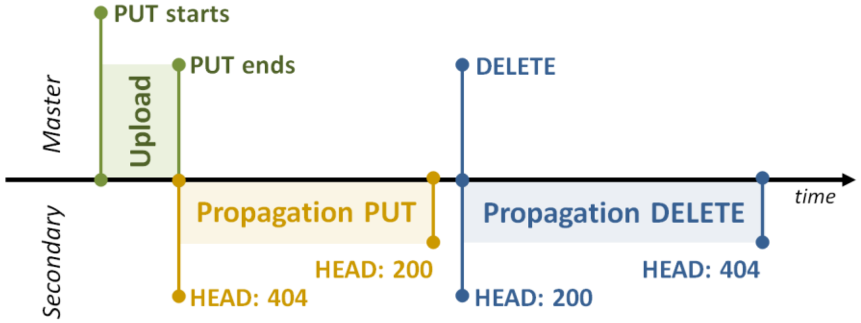
1. Sync may lag behind and struggles to recover
   - We wrote 1M objects to the master zone, while secondary was shut-off
   - It took ~1 day to sync with no other load on the clusters

	PUT	Propagation PUT	Propagation DELETE
Mean	1.15	7.75	10.80
Stdev	0.01	2.70	2.00

2. Intrinsic inter-zone replication delay:
   - Full mirror mode implies eventual consistency
   - Secondary zone may not have most-recent objects



	Mean	Min	Max	Range
🟩 Upload (PUT)	1.15 s	1.14 s	1.18 s	33.5 ms
🟨 Propagation on upload (HEAD on 2nd region after PUT)	7.75 s	243 ms	15.8 s	15.6 s
🟦 Propagation on removal (HEAD on 2nd region after DELETE)	10.8 s	4.62 s	17.2 s	12.6 s



2. S3 Objects: Immutable Backups

- **What for:** DR – Backups Store

- **Immutable S3 Objects with Retention Policies**
  - Versioning: PUTs on existing objects preserve existing data as previous object version (w/ versionID)
  - Object Locks: Prevent deletions to objects (and versions) for a retention period
  - Retention: Predefined (user/admin choice) to defer deletions

- **Archive Zone**
  - Solves the problem of having a global zone archiving all objects versions
  - Understands bucket versioning with no write amplification
  - Likely on slower, cheaper media
    - Not the case yet – Shingled disks or tape in the future?

3

File System

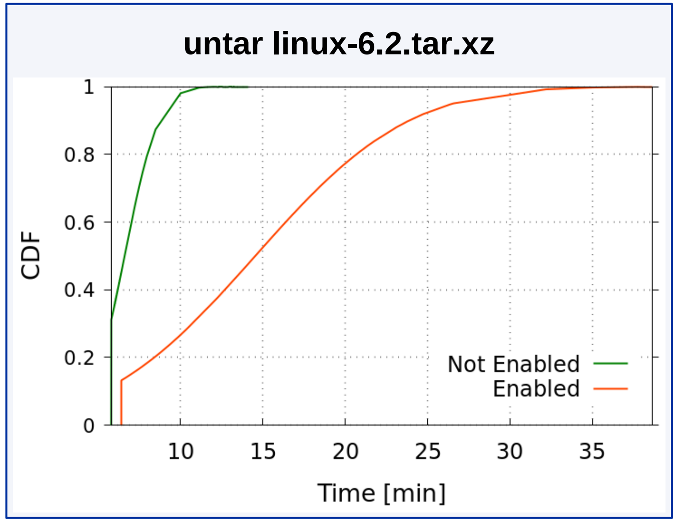
3. CephFS: Snapshots (and Mirroring)

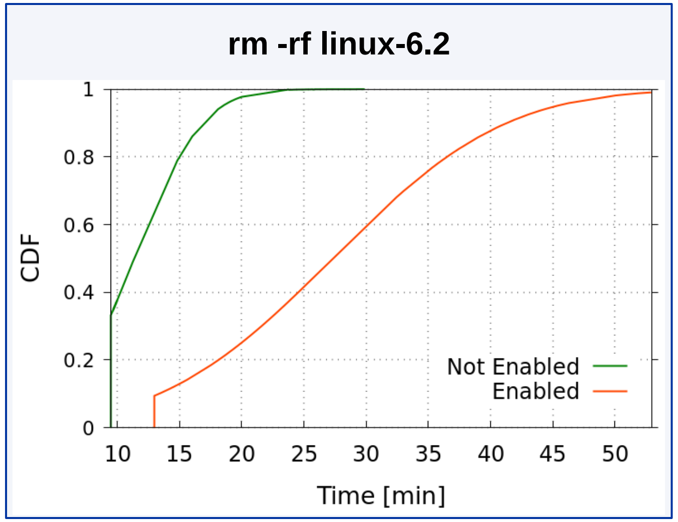
- **What for:** BC/DR – Rollback, Backup

- **Immutable point-in-time view of a file system**
  - Snapshots can be triggered by users, or automated by admin
  - Existing snapshots accessible at `.snap` directory
  - Creation is fast: Lazy flush, copy-on-write

- **Severe impact on performance**
  - Tested some metadata intensive workloads (Pacific 16.2.9)
  - Done in a 10-level deep directory tree containing 100 empty or sparse directories

  - Problem seems localized in the Metadata server, kept busy tracking ancestors
  - Trying to work-around by isolating FS with snaps on dedicated MDSs
    - Helpless if everyone wants snapshots…



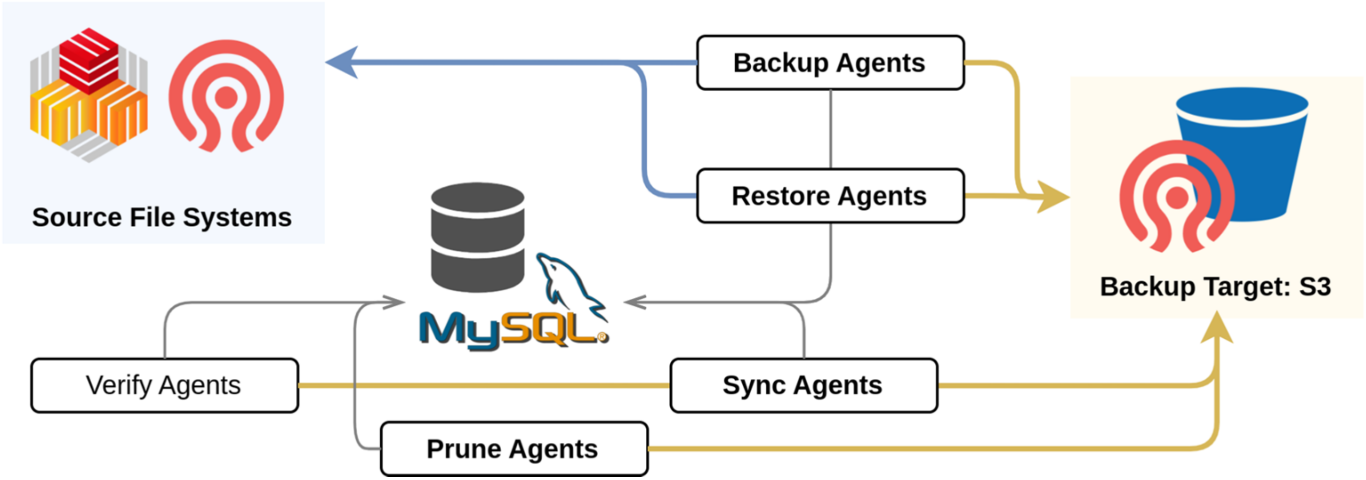


3. CephFS: Restic Backups at Scale with cback

- **What for:** DR – Backup & Restore

- **Backup orchestration tool for File Systems**
  - Based on Restic, with the addition of horizontally-scalable agents
  - Used to backup EOS/CERNBox and (some) CephFS

  - Source: Any mounted file system
  - Destination: Ceph S3

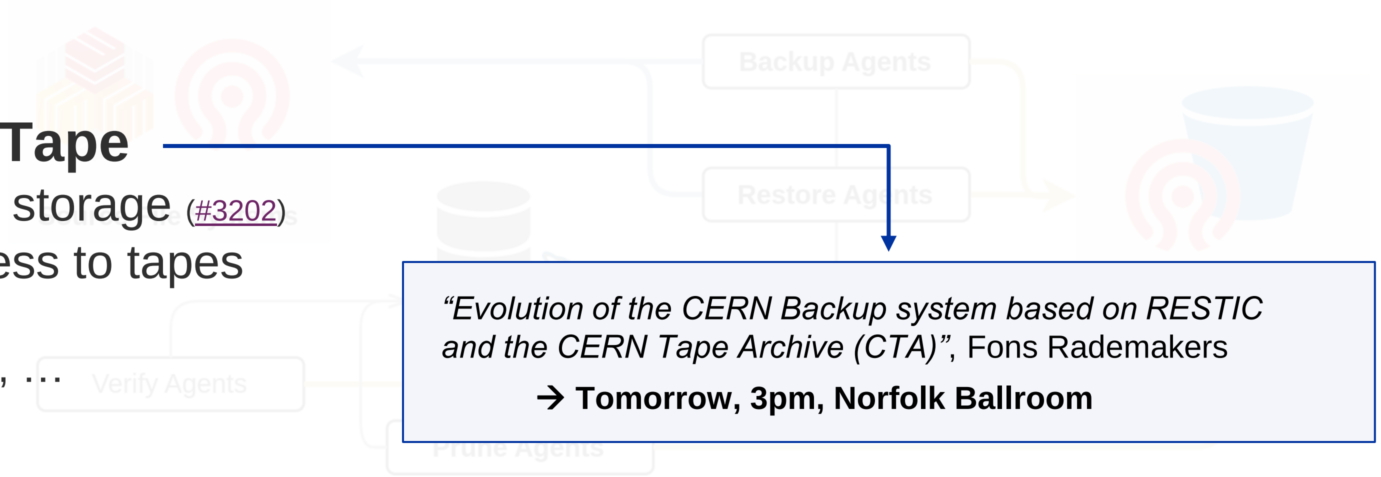


3. CephFS: Restic Backups at Scale with `cback`

- **What for:** DR – Backup & Restore

- **Backup orchestration tool for File Systems**
  - Based on Restic, with the addition of horizontally-scalable agents
  - Used to backup EOS/CERNBox and (some) CephFS
  - Source: Any mounted file system
  - Destination: Ceph S3

- **Next challenge: Write backups to Tape**
  - Restic expects (meta)data to be on hot storage (#3202)
  - Improvements needed to optimize access to tapes
    - Object sizes, access frequency, fragmentation over multiple tapes, …



“Evolution of the CERN Backup system based on RESTIC and the CERN Tape Archive (CTA)”, Fons Rademakers

→ **Tomorrow, 3pm, Norfolk Ballroom**

Conclusions

1. **There is no catch-all solution**
   - BC and DR are different concepts with different goals, and require different technical solutions – Active/Active vs Backup&Restore
   - Block, Object, and File System come with different features for BC/DR
2. **Feature maturity greatly differs**
   - Snapshots for CephFS have severe performance implications, RBD backups works out the box nicely.
   - S3 multisite “works” with some limitations and increased operational complexity

- **Work continues:**
  - Finalize cross-cluster RBD backups and prepare for production deployment
  - Use **`cback`** for CephFS backups more widely

Norfolk, Virginia, USA • **May 8-13, 2023**

CHEP 2023

Computing in High Energy & Nuclear Physics

Thank you!

**Enabling Storage Business Continuity
and Disaster Recovery with Ceph distributed storage**

Enrico Bocchi
enrico.bocchi@cern.ch

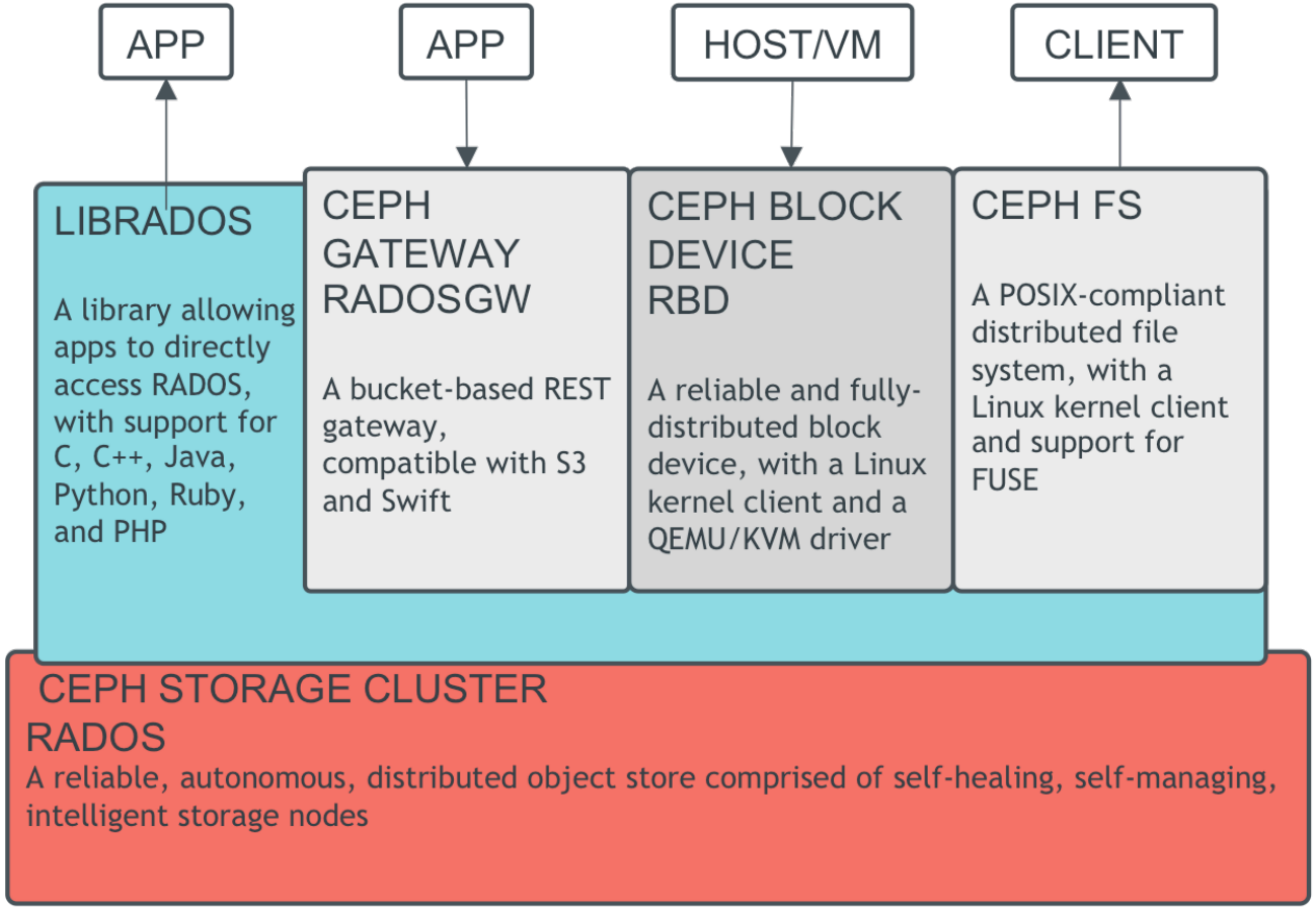
Backup

What is Ceph?

- **Free and Open storage software**
  - **RBD**: Virtual Block device
  - **RADOSGW**: S3-compatible storage
  - **CephFS**: Scalable distributed filesystem

- **Reliable and Durable**
  - Favor consistency and correctness over performance (or availability)
  - No single point of failure
  - Replication or EC

- **Scalable**
  - Online add/remove storage, software upgrades
  - Single-cluster or multi-cluster federation


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
CEPH GATEWAY RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
CEPH BLOCK DEVICE 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
CEPH STORAGE CLUSTER
RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Ceph at CERN

Application		Size (raw)	Version
RBD (OpenStack Cinder/Glance, `krbd`)	*Production, HDDs*	24.5 PiB	Pacific
	Production, full-flash EC 4+2	643 TiB	Pacific
CephFS (OpenStack Manila – K8s/OKD PVs, HPC)	*Production, HDDs*	7.9 PiB	Pacific
	Production, full-flash	782 TiB	Pacific
	Hyperconverged (HVs with flash storage)	892 TiB	Octopus
CERN Tape Archive (CTA)	*Tape DB and Disk Buffer*	235 TiB	Octopus
RGW (S3 + SWIFT)	*Production (4+2 EC)*	4.1 PiB	Octopus
S3, RBD: Backup to 2nd Location	*Production (4+2 EC, 3 replicas)*	25 PiB	Octopus

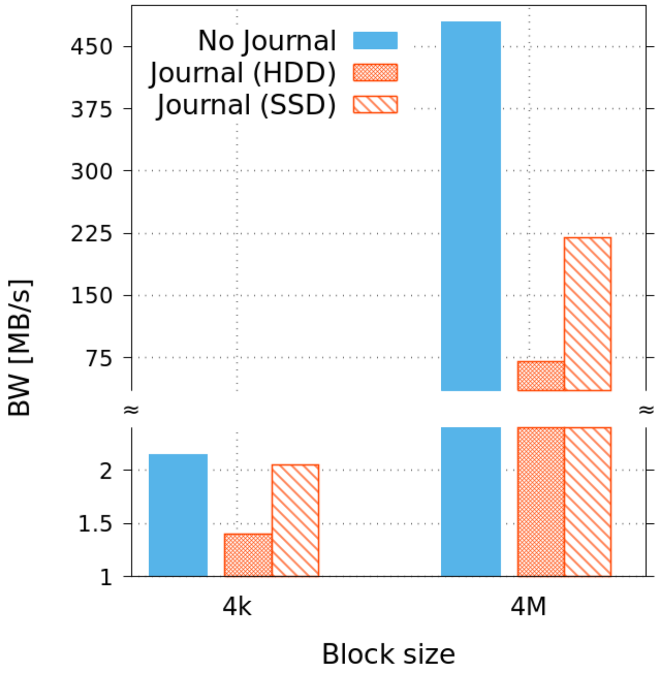
1. RBD: Mirroring

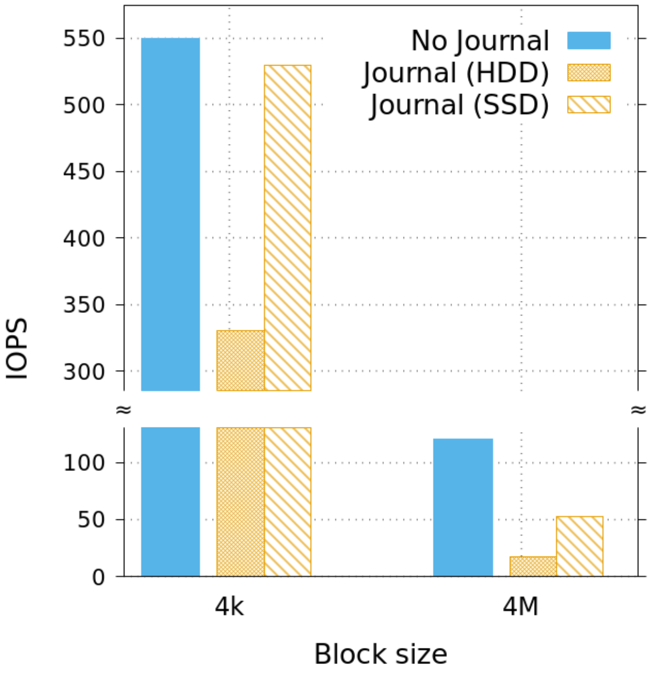
- **What for:** BC/DR – Active/Passive Setup

1. **Managed by `rbd-mirror` daemon**
    - Reads state of RBD images from source to replay asynchronously on target
    - RBD client writes to image and journal
    - Severe impact on client performance
    - Replays are slow: ~30 MB/s (but scale well with number of images)
    - Risk of lagging behind:
        - Replicas get out-of-date
        - RBD journal not trimmed

Testbed

- 6 bare-metal servers:
    - Data on 60 HDDs (+ blockdb on SSD)
    - RBD journal on HDDs or SSD
- 5 clients (`librbd`) running multiple `fio`, random write





1. RBD: Mirroring

2. **Snapshot-based Mirroring**
   - Allows for point-in-time replication
   - Image snapshot diff exported from main cluster, then imported to mirror target

   - Performance impact only related to:
     - Snapshot trimming and replay workload
     - RBD client not involved in replication
   - Replays are fast: ~200 MB/s per image

   - Several improvements and fixes in Ceph (GitHub)
   - Not supported (yet) by OpenStack (OpenDev)

1. RBD: Mirroring

- **What for:** DR – Backup & Restore

- **Mirroring based on Snapshots:**
  - Allows for point-in-time replication
  - Image snapshot diff exported from main cluster, then imported to mirror target

  - Performance impact only related to:
    - Snapshot trimming and replay workload
    - RBD client not involved in replication
  - Replays are fast: ~200 MB/s per image

  - Several improvements and fixes in Ceph (GitHub)
  - Not supported (yet) by OpenStack (OpenDev)

2. S3 Objects: Sync to External Clouds

- **Two modules available, sadly almost abandoned**

1. **Cloud Transition**
    - Potential use case: Transition to a remote site for cold-media backups
    - Requires local zone modification + storage class creation
    - Lifecycle policies on a per bucket policy, no site-wide policy
    - Limitation – Currently single account key for remote site

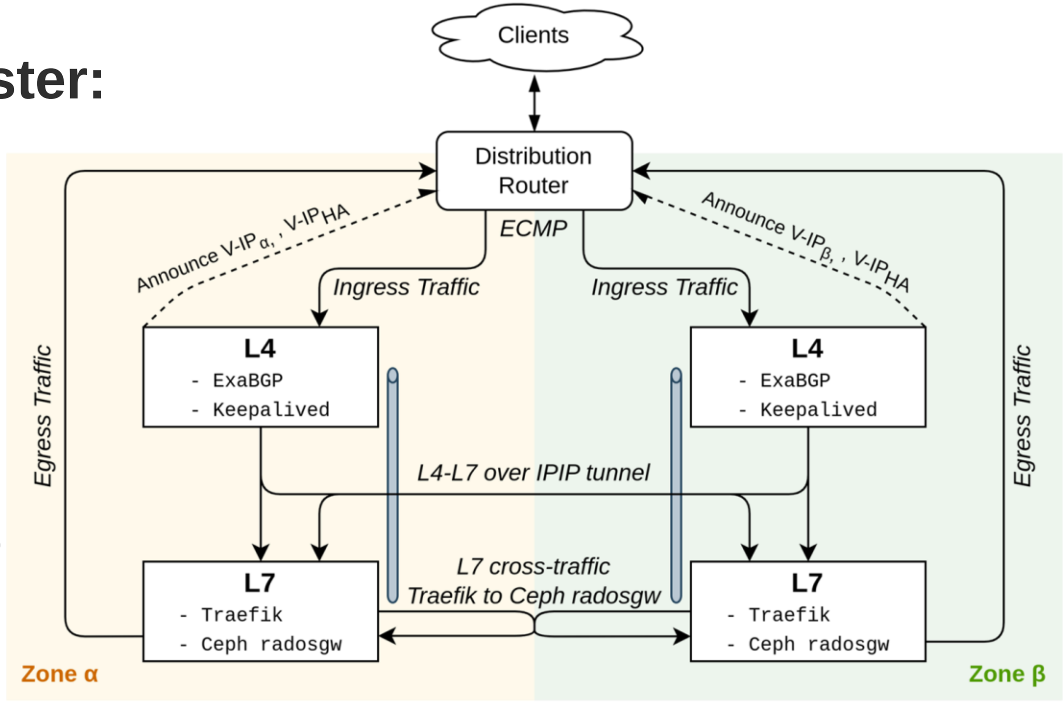
2. **Cloud Sync Module**
    - Potential use case: Keep copy of (very) critical data on cloud that can be used by local compute
    - Requires separate zone which acts as a pipe to move data
    - Limitation – Saw several crashes on misconfiguration; Requires effort to bring to production grade

2. S3 Objects: BGP Load Balancing

- **DNS load balancing has several limitations:**
  - Reacting to change hints for low TTL (recursive queries may hit a minimum TTL)
  - Client behavior is implementation-specific (libraries, OSes, caches, …)

- **Expose 1 Virtual-IP for the whole multisite cluster:**
  - Routers forward traffic to L4s with 5-tuple hashing
  - L4 balancers:
    - Peer with routers announcing one V-IP (`ExaBGP`)
    - Forward to L7s with consistent hashing (Maglev) over IPIP
  - L7 balancers:
    - Run Traefik frontend and Ceph radosgw
    - Answer to clients through direct return paths with routers

  - Allows directing clients to the closest zone (lower metric)
    - Or fallback to other zone if preferred is unavailable
  - Does not help with replication delay between zones


Clients
Distribution Router
ECMP
Announce V-IP$_\alpha$, V-IP$_{HA}$
Announce V-IP$_\beta$, V-IP$_{HA}$
Ingress Traffic
Ingress Traffic
Egress Traffic
Egress Traffic
L4
- ExaBGP
- Keepalived
L4
- ExaBGP
- Keepalived
L4-L7 over IPIP tunnel
L7 cross-traffic
Traefik to Ceph radosgw
L7
- Traefik
- Ceph radosgw
L7
- Traefik
- Ceph radosgw
Zone α
Zone β