Experience with CVMFS and Ceph/S3

Status Update

Enrico Bocchi
CERN IT – Storage

CernVM Virtual Workshop 2021
1-2 February 2021, NIKHEF (remote)

CVMFS at CERN

CVMFS

- Stratum 0 servers
  - Release Managers
  - Gateways
- Stratum 1 servers
  - Replica server
  - Front-end caches
- Backup

OurProxy

- Site caches for clients

Clients

- Software environments on LXPLUS
- Batch Farm
- Experiments' online farms
- SWAN Jupyter Notebooks
- Hadoop clusters
- Scientists' laptops

Outline

- Migration of Stratum 0 storage to S3
  - S3 service at CERN
  - S3 tuning for CVMFS
  - Benefits of S3 Storage
- Content distribution to Clients
  - Stratum 1 Replica Server
  - Dedicated sets of caches for major repositories
  - “Pass-through” repositories
- Conclusions and Future Outlook

Migration to S3

- S3 service at CERN
- S3 tuning for CVMFS
- Benefits of S3 Storage

S3 Storage at CERN

- Production service since 2018: s3.cern.ch
  - Started in 2016 as ATLAS event service
  - Default storage for new repositories since Q4 2018

- Single-region radosgw cluster
  - 5.8 PB raw capacity, 810 TB raw used, 358.04 M objects
  - 4+2 erasure coding for data, 3x replication for bucket indexes
  - Available from OpenStack as object storage for projects

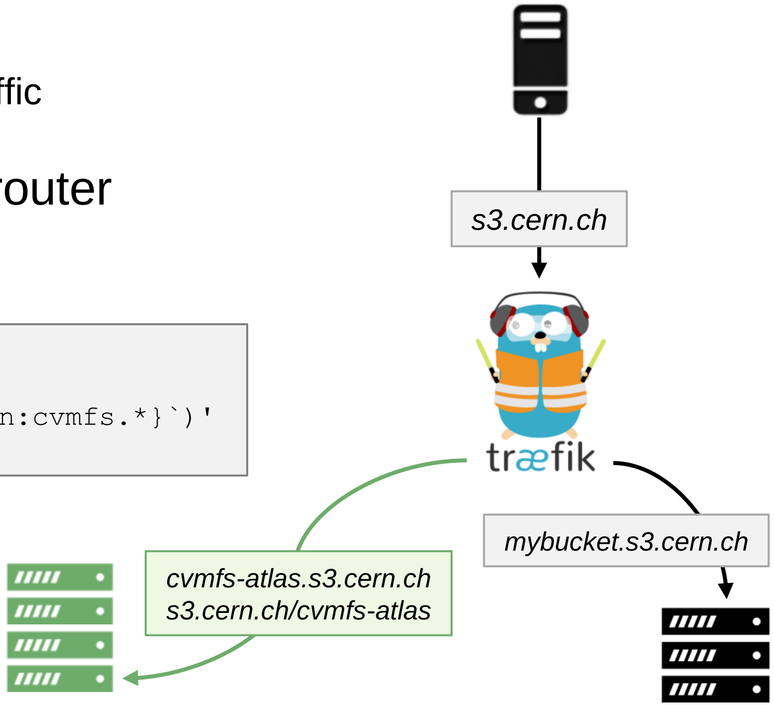
- 2nd S3 cluster in Prévessin network hub: s3-fr-prevessin-1.cern.ch
  - This is not a second region
  - Used for backups and disaster recovery

S3 Storage at CERN

- s3.cern.ch is
  - 10 load-balanced IPs with Traefik
  - 16 active radosgws, 4 dedicated for CVMFS traffic

- Traefik as frontend and application-level router
  - TLS termination, radosgw health-check
  - Dynamic routing based on path and virtual host

```
routers:
  s3_cvmfs:
    rule: 'PathPrefix(`/cvmfs`) || HostRegexp(`{pattern:cvmfs.*}`)'
    service: cvmfs
```



Migration of existing repositories

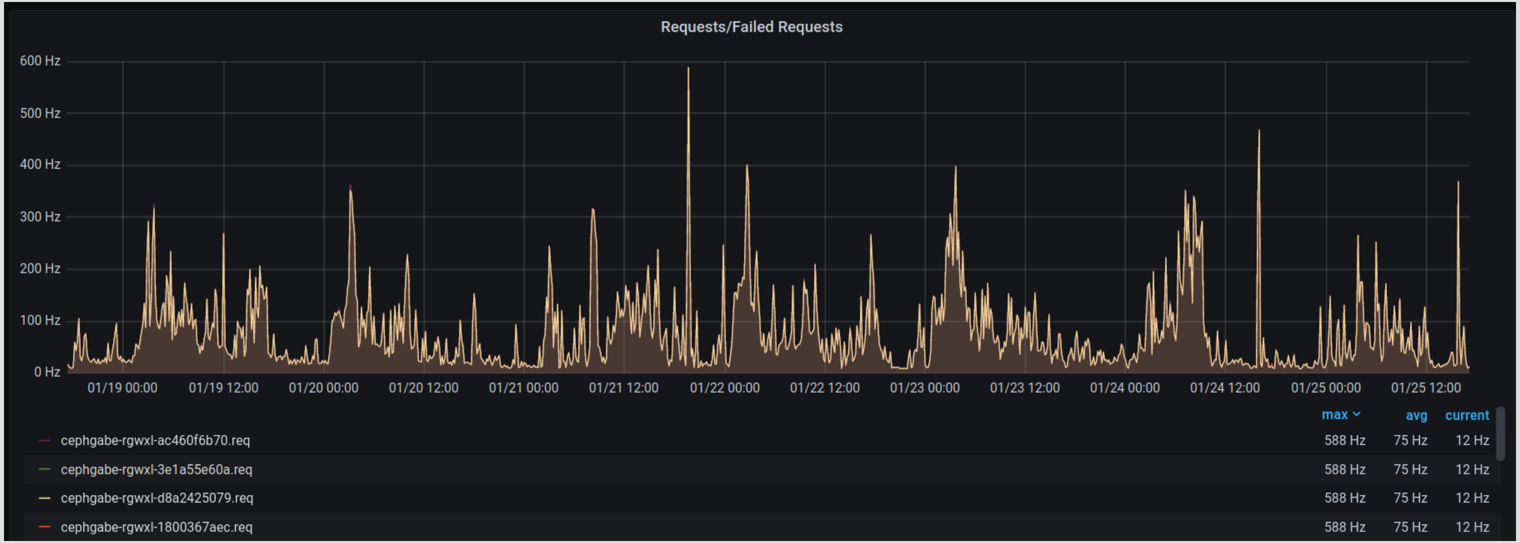
- In 2020, 34 repositories migrated to S3
  - 740 M objects (64% of total objects), 37.99 TB (54% of total bytes) at the time of migration
  - One S3 user per repository, one bucket per repository
  - Many critical repositories from major LHC experiments (atlas.cern.ch, lhcb.cern.ch, …)

- Migration via `cvmfs_server snapshot <repo>`
  - Get rid of unreferenced objects – Implicit garbage collection

- Reduce impact of migration
  - Transactions are not allowed during intervention
  - Initial replication to S3 prepared days before the actual migration
  - Intervention for final replication and minor re-configuration
  - Where needed, upgrade release manager to CC7

S3 Storage for CVMFS

- CVMFS is the top S3 user for number of IOPS
  - Average number IOPS is moderate (~300 Hz)
  - Can be very spiky – Observed peaks over 4KHz



	max	avg	current
cephgabe-rgwxl-ac460f6b70.req	588 Hz	75 Hz	12 Hz
cephgabe-rgwxl-3e1a55e60a.req	588 Hz	75 Hz	12 Hz
cephgabe-rgwxl-d8a2425079.req	588 Hz	75 Hz	12 Hz
cephgabe-rgwxl-1800367aec.req	588 Hz	75 Hz	12 Hz

S3 Tuning for CVMFS

- IOPS: Request throttling not really successful
  - Traefik can return ‘429 – Too Many requests’
  - radosgws (and AWS’ S3) can limit with ‘503 – Slow Down’

- Result: 4 dedicated radosgws for CVMFS traffic

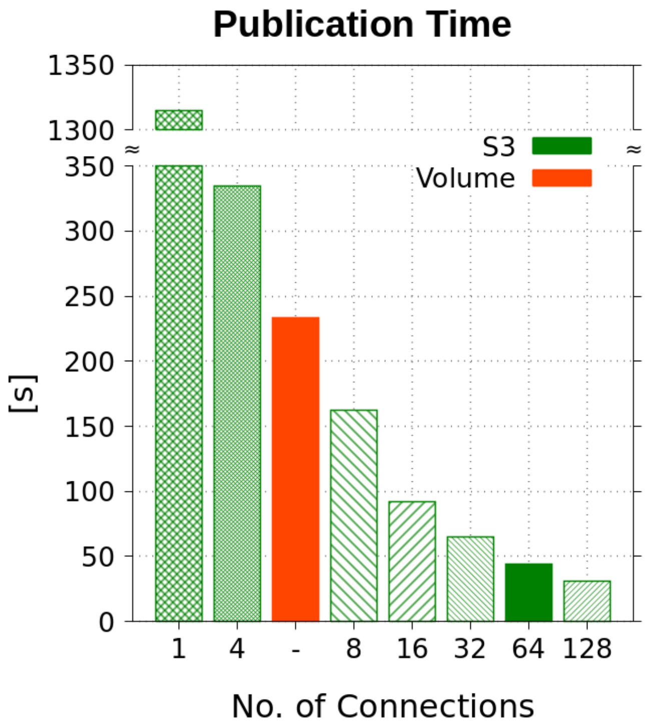
- No. of Objects: Bucket indexes sharding
  - radosgws maintain an index with metadata
  - Index is typically sharded (default 32 shards)
  - Need to find a good balance between number of indexes and index size

- CVMFS implications
  - Ever-growing repos require offline re-sharding
  - Auto-resharding not (yet) enabled

Repository	Volume [GB]	Objects [M]
cvmfs-cms	7904.41	**29.38**
cvmfs-lhcb	1108.25	**18.92**
cvmfs-sft	1959.69	**15.06**
cvmfs-atlas	1891.48	**11.64**
cvmfs-na62	30.74	**9.17**
cvmfs-cms-ib	383.5	**5.77**
cvmfs-atlas-nightlies	1792.64	**5.32**
cvmfs-ams	2657.75	**5.05**

Benefits of S3 Storage

- Improvement in performance
  - ➢ Publication time benchmarking
    - ✓ Sample workload: 250k files, 4 kB each
    - ✓ Files are organized in 250 folders
    - ✓ Each folder has a dedicated CVMFS catalog
    - ✓ Time is full publication chain through cvmfs_server
  - ➢ Default number of parallel connections: 64
  - ➢ S3 with parallel uploads outperforms volume storage
  - ➢ Publication on S3 is 5x faster



Benefits of S3 Storage

- Improvement in service operations

	SLC6 + Volume	CC7 + S3
Authoritative Storage	Volume with ZFS (zfs-kmod required)	S3 HTTP endpoint
CVMFS Union FS	AUFS-enabled custom kernel	OverlayFS
Quota Management	Intervention required (detach, expand, zfs magic)	Online (one line cmd on radosgw-admin)
Release Manager Failover	Detach storage volume and sanity check	Spawn new VM in minutes
HTTP Access	Single VM with httpd	Redundant S3 cluster 4 RGWs for CVMFS traffic

2

Content Distribution to Clients

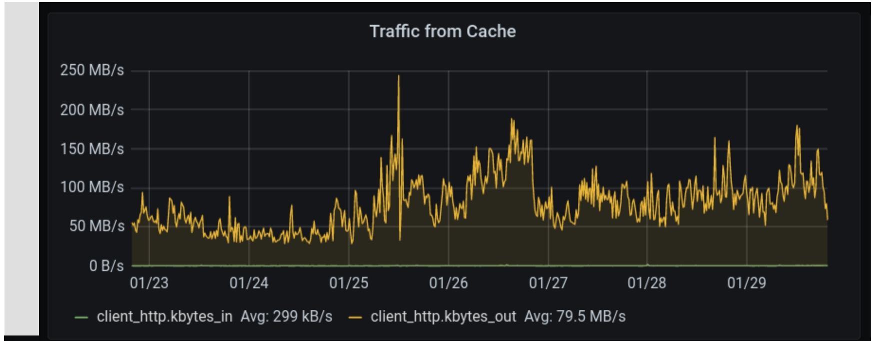
- Stratum 1 Replica Server
- Dedicated caches for major repositories
- “Pass-through” repositories

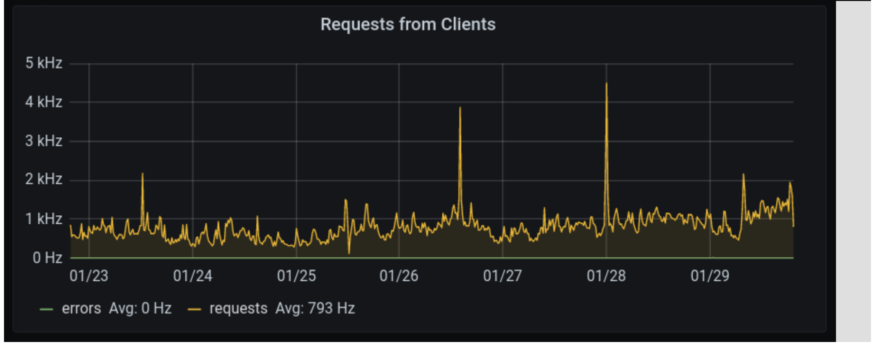
Stratum 1 at CERN

- Replicate from S3 to Stratum 1
  - cvmfs_server snapshot

- Backend
  - Physical server with 24x6 TB disks
  - Single large ZFS volume of 130 TB
  - Serving frontend:
    ~20 MB/s, ~100 Hz

- Frontend – cvmfs-stratum-one.cern.ch
  - 4 VMs with ~2.2 TB cache on SSD
  - frontier-squid as reverse proxy
  - Serving site caches + clients:
    ~80 MB/s, ~800 Hz





Dedicated Site Caches for Major Repositories

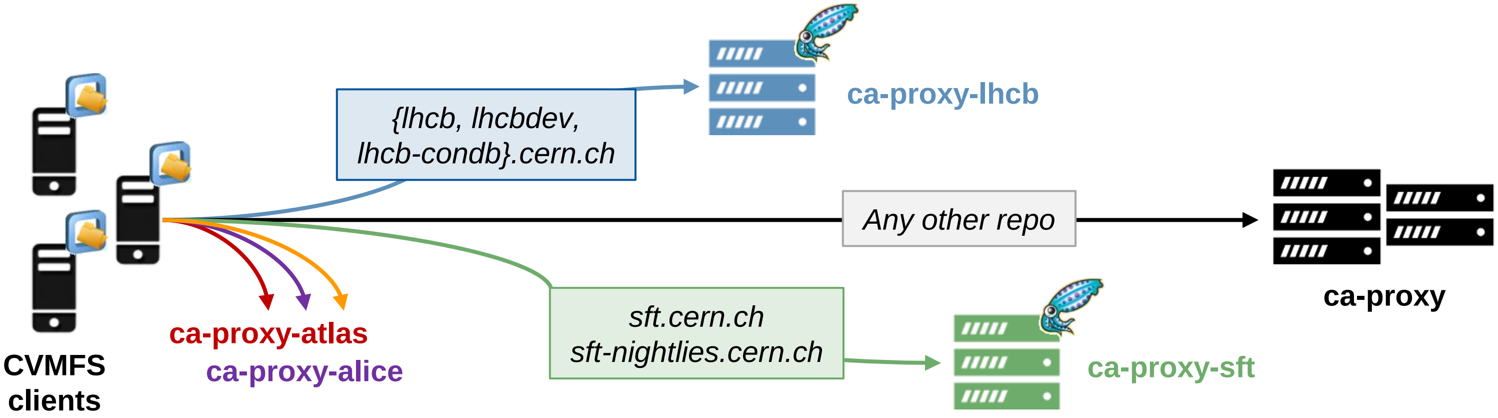
- Starting point: One pool (ca-proxy.cern.ch) of 10 caches serving all repos
  - VMs with 160GB cache (on SSD), 10Gbps network
  - Squid caching software as forward proxy

- Problem 1: Caches get inefficient (requests/traffic hit rates decrease)
  - Cache do not coordinate / peer. They all tend to cache the same items
  - Size of the repositories constantly increases, size of caches does not

- Problem 2: Cross-repositories interference
  - One repository “abusing” caches degrades the access to all the other repositories (similar to DDoS)
  - Difficult to apply effective countermeasures when detected (traffic shaping?)
  - Several incidents in the past caused by atypical reconstruction jobs fetching dormant files

Dedicated Site Caches for Major Repositories

- Goal: Reduce interference across repositories and improve cache efficiency
- Result: Dedicated caches for groups of repositories
  - 5 sub-pools of caches for main LHC experiments (ca-proxy-alice, ca-proxy-atlas, …) + 1 for SFT
  - Several CNAMEs (e.g., ca-proxy-compass, ca-proxy-ams, …) to steer traffic in case they cause overloads
  - 1 pool of general caches remains for all other repos (ca-proxy.cern.ch)
  - All caches updated to frontier-squid 4 series

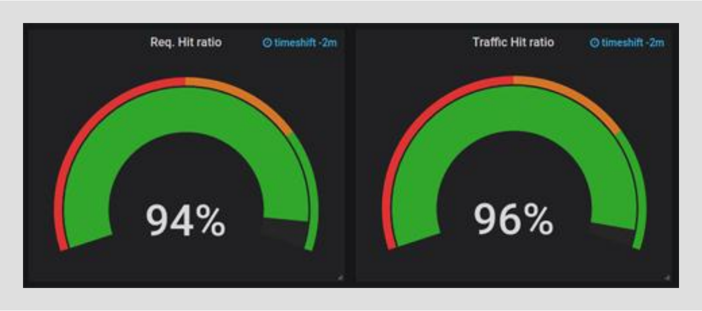


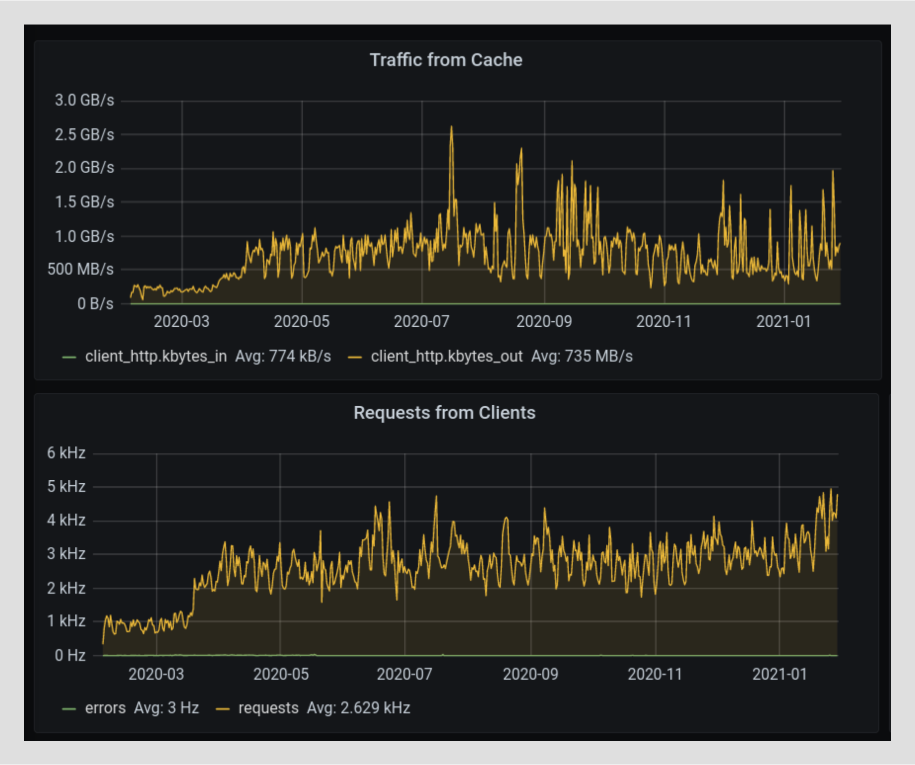
Traffic Served by Dedicated Caches

- One year of dedicated caches
  - 5 sub-pools – 3 VMs per pool
  - Different AZs, ToR switches, routes, …
  - ~380 GB cache space per pool

- No cache trashing // overloads

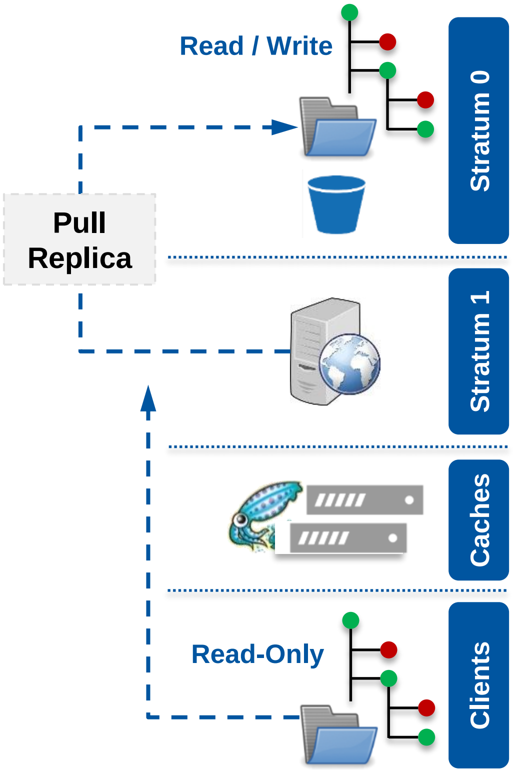
- Cache efficiency is *very* high
  - >90% both request- and byte-wise





Pass-Through Repositories

- Typically, clients read from Stratum 1 (through caches)
  - A (very small) replication delay exists between Stratum 0 and 1
  - Stratum 1 might lag behind when garbage collecting

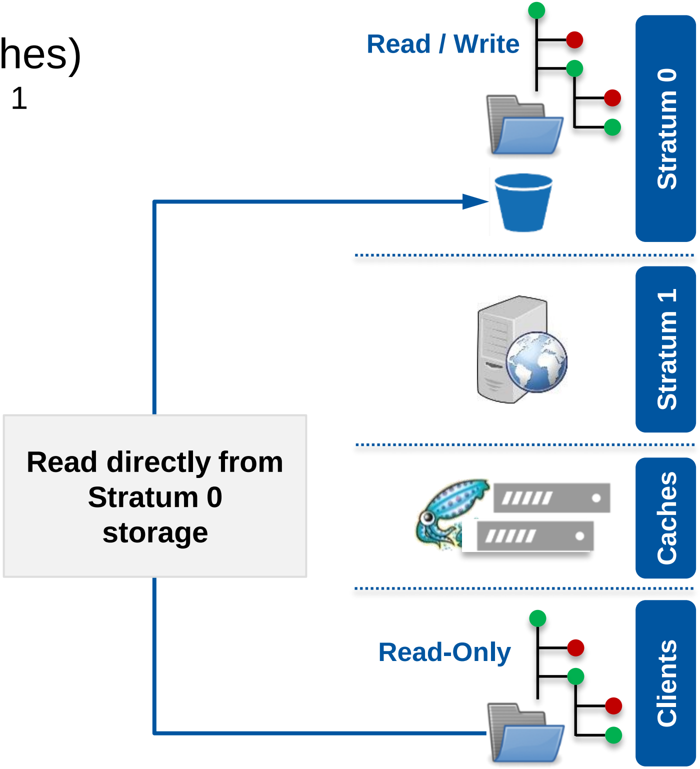


Pass-Through Repositories

- Typically, clients read from Stratum 1 (through caches)
  - A (very small) replication delay exists between Stratum 0 and 1
  - Stratum 1 might lag behind when garbage collecting

- S3 enables to read directly from Stratum 0 storage
  - No replication delay
  - Garbage collection is not blocking for reads

- Relevant for lhcbdev.cern.ch
  - Nightly releases repository – High churn rate
  - Regularly (and heavily) garbage collected
  - GC on the Stratum 1 might inhibit replication for too long



3

Conclusions and Future Outlook

Conclusions

- CVMFS at CERN evolved with new infrastructure and components
  - Stratum 0 storage fully based on S3
  - Dedicated caches for major repositories
  - CVMFS Gateway entered production for high-performance use case

- Improved service for repository owners
  - Simplified management of the infrastructure
  - Faster publication with parallel transactions and Gateway
  - More resilient content distribution to clients

Future Outlook

1. Replication and Garbage Collection on Stratum 1
   - Stratum 1 might be unable to snapshot (for too long) when garbage collecting
   - Pass-through or avoid GC and regularly make new snapshots from scratch

2. Distribution of container images
   - Could generate a relevant amount of traffic on the infrastructure
   - Deployment of dedicated caches is an option
   - User uptake unclear at the moment

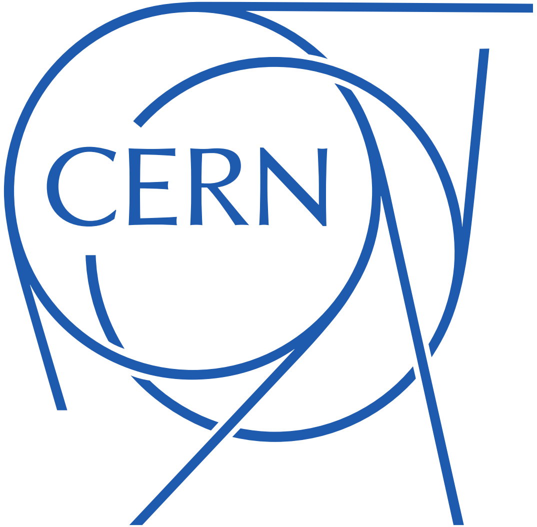
3. Bucket index sharding on S3
   - Improvements coming from upstream with new releases
   - Manual interventions (and downtime) are very infrequent

CernVM
File system

Thank you!

Questions? || Comments?

Enrico Bocchi
enrico.bocchi@cern.ch


CERN

Backup

CVMFS Main Content Types

1. Production Software
   - Most mature use case
   - E.g., /cvmfs/atlas.cern.ch

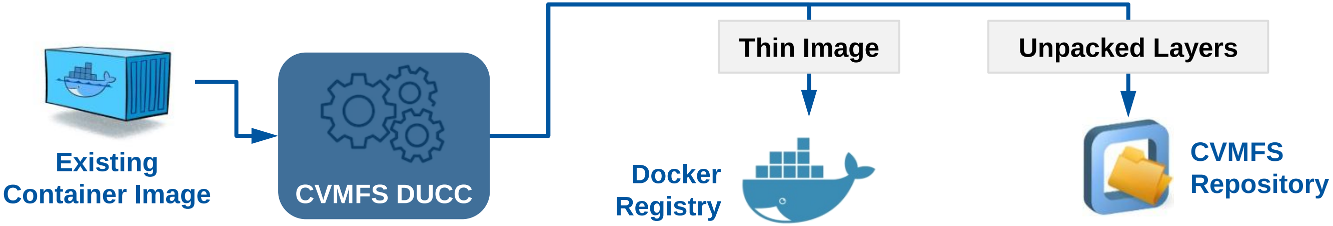
2. Auxiliary Datasets
   - Benefits from internal versioning
   - E.g., /cvmfs/alice-condb.cern.ch

3. Integration Builds
   - High churn, requires regular garbage collection
   - E.g., /cvmfs/lhcbdev.cern.ch

4. Container Layers Ingestion
   - Benefit from de-duplication and on-demand caching
   - unpacked.cern.ch

CVMFS for Container Layers Distribution

- Server: Ingestion via DUCC
  - Publishes container images in their extracted form on CVMFS
  - Generates and uploads the *Thin Image* on Docker registries



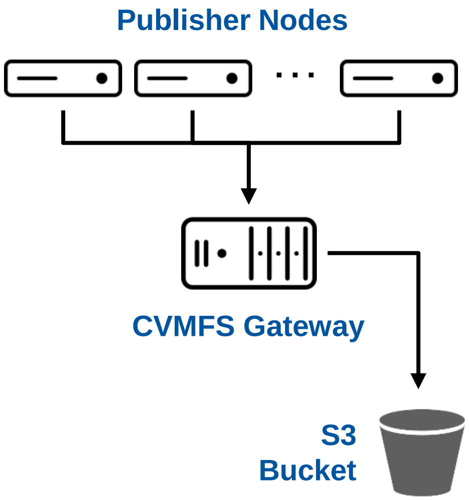
- Client: Container Runtime Integration
  - No need to download and extract images locally
  - Native support for Singularity and runc (flat runtime)
  - *Graph Driver* for Docker, containerd/k8s, Podman (layered runtime)

2

CVMFS Gateway

CVMFS Gateway

- Stateful component allowing for concurrent publications
  - Issues time-limited leases for specific sub-paths
  - Provides API to coordinate across publishers
- Exclusive write access to S3 storage
  - Publishers ship object packs to Gateway
  - Gateway commits changes to storage and updates repo manifest
- Operational limitations
  - GC from Gateway only
    - ➔ Progress reporting implemented
  - Warnings on catalog sizes trigger publication errors
    - ➔ Enable autocatalogs

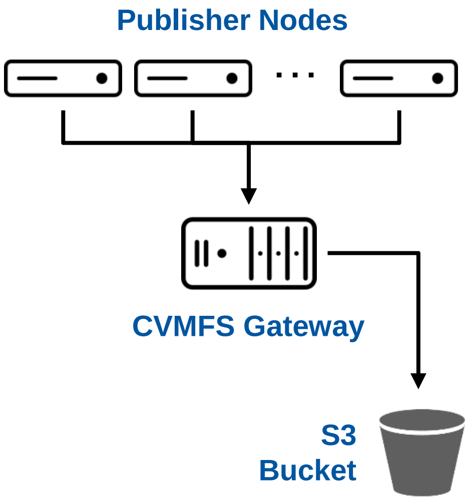


CVMFS Gateway

- Running in production since January 2019
  - Traditional repository for software publication
  - Multi-tenant repository with RW access to different subpaths

- Q4 2020 deployed for high-performance use case
  - Publication of nightly builds
  - Reduced time to publish all builds (and/or publish more builds)

- To what extent Gateway allows linear scalability?
  - Production deployment with few publishers looks very promising
  - Large scale tests not performed (yet)


Publisher Nodes
. . .
CVMFS Gateway
S3
Bucket