CERN

CVMFS

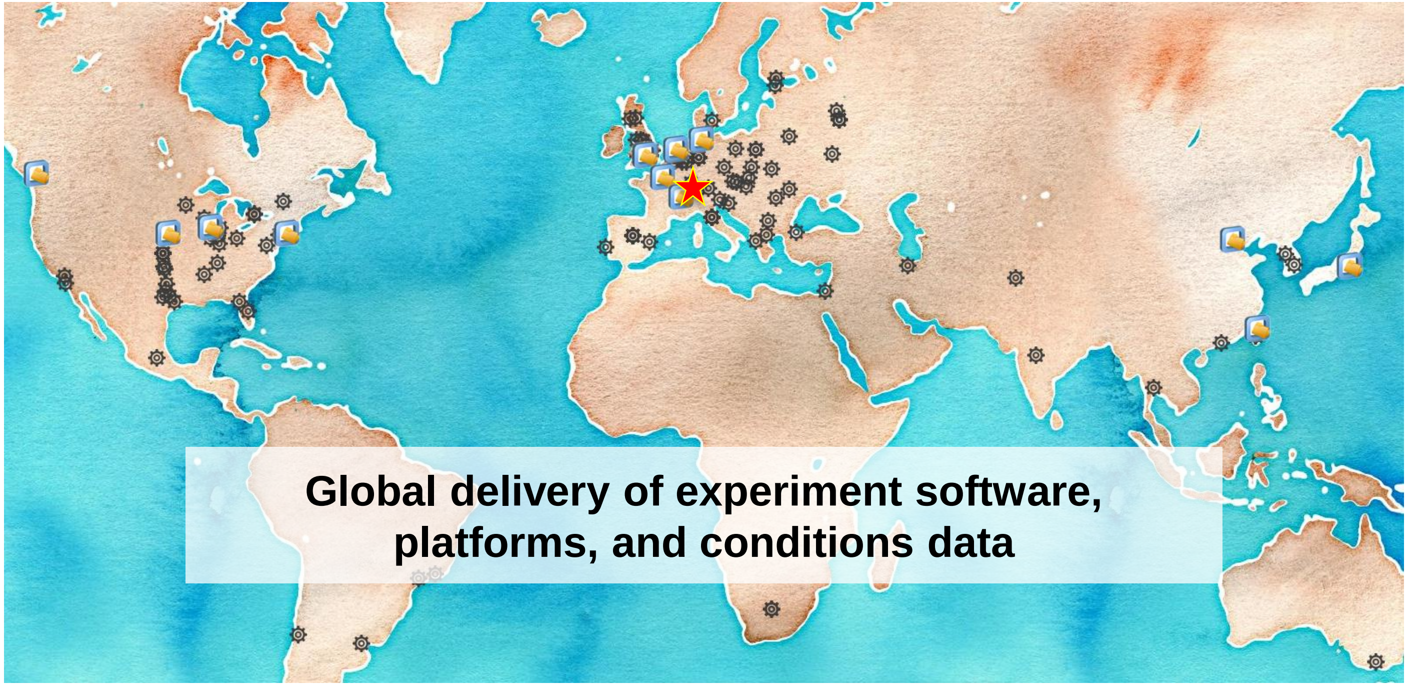
Service Evolution and Infrastructure Improvements

Enrico Bocchi

CERN, IT-Storage

HEPiX Online, October 2020

CVMFS in a Nutshell

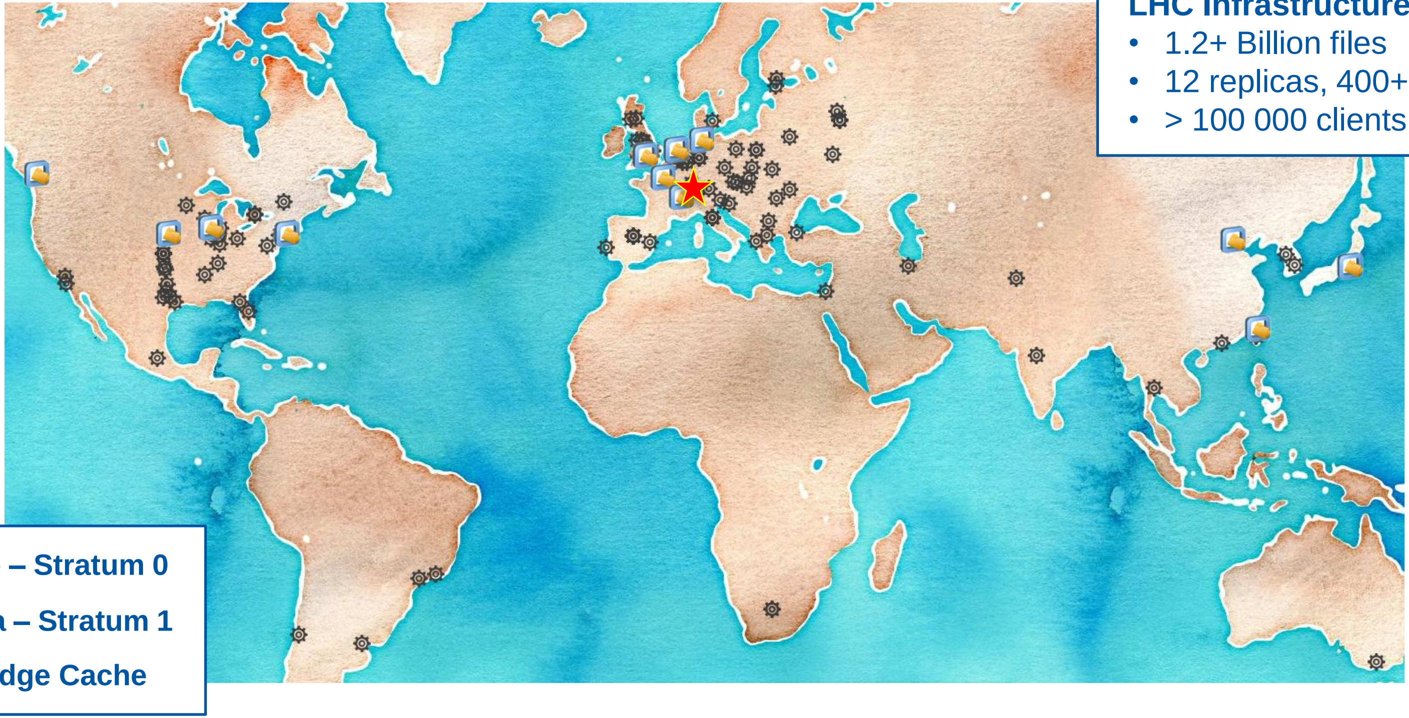


Global delivery of experiment software, platforms, and conditions data

CVMFS in a Nutshell

LHC Infrastructure

- 1.2+ Billion files
- 12 replicas, 400+ caches
- > 100 000 clients



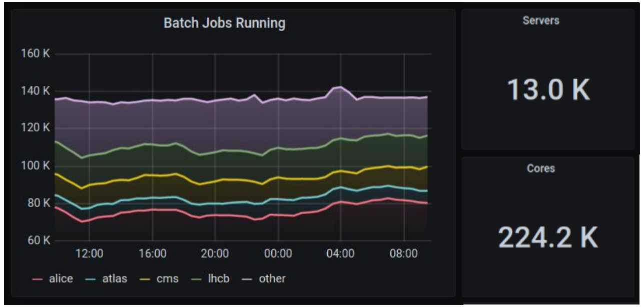
★ **Source – Stratum 0**

▣ **Replica – Stratum 1**

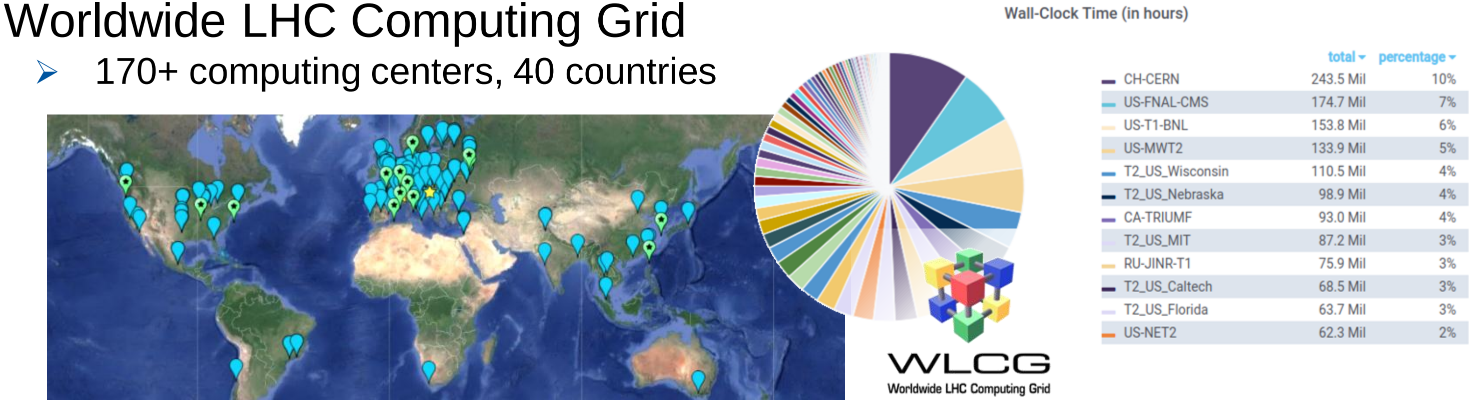
⚙ **Site / Edge Cache**

CVMFS in a Nutshell

- Ubiquitous CVMFS client at CERN
  - Batch jobs, Hadoop clusters
  - Experiments’ online farms
  - SWAN Jupyter Notebooks
  - Scientists’ laptops



- Worldwide LHC Computing Grid
  - 170+ computing centers, 40 countries



Wall-Clock Time (in hours)

	total ▾	percentage ▾
CH-CERN	243.5 Mil	10%
US-FNAL-CMS	174.7 Mil	7%
US-T1-BNL	153.8 Mil	6%
US-MWT2	133.9 Mil	5%
T2_US_Wisconsin	110.5 Mil	4%
T2_US_Nebraska	98.9 Mil	4%
CA-TRIUMF	93.0 Mil	4%
T2_US_MIT	87.2 Mil	3%
RU-JINR-T1	75.9 Mil	3%
T2_US_Caltech	68.5 Mil	3%
T2_US_Florida	63.7 Mil	3%
US-NET2	62.3 Mil	2%

Outline

1. CVMFS for Container Layers Ingestion and Distribution
2. Infrastructure Improvements
   - S3 as Stratum 0s Storage
   - Dedicated Caches for Content Delivery
3. Conclusions

New CVMFS Capabilities

Container Layers Ingestion and Distribution

1.1 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`

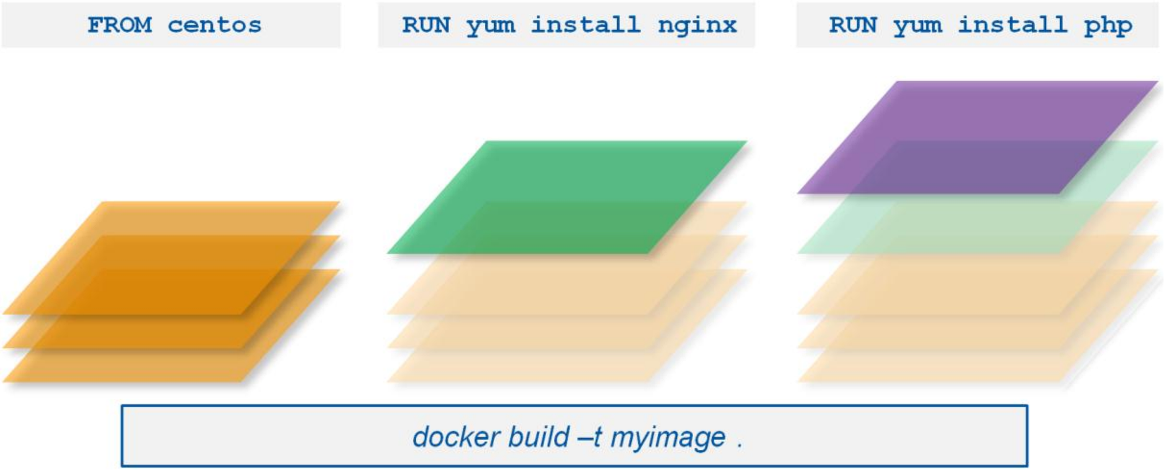
Container Layers

- Ingestion and Distribution of Container Images
- Benefit from de-duplication
  and on-demand caching
- e.g., `/cvmfs/unpacked.cern.ch`

J. Blomer – CVMFS for Containers
Thu 15 Oct, 15:00
https://indico.egi.eu/event/5251/

1.1 CVMFS for Containers

- Container images are the product of several layers
  - Layers are TAR files
  - Need to be downloaded and extracted

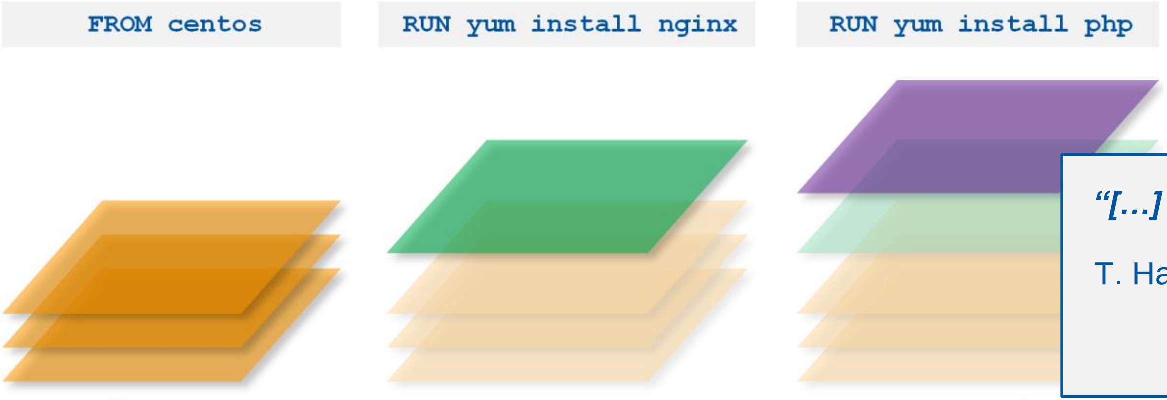


```
FROM centos
RUN yum install nginx
RUN yum install php
```

docker build –t myimage .

1.1 CVMFS for Containers

```
[root@ThinkPad-X1]# docker history myimage
IMAGE               CREATED             CREATED BY                                      SIZE
75cc2375258a        4 seconds ago       /bin/sh -c yum -y install php                   66.9MB
e779b8a4024f        9 seconds ago       /bin/sh -c yum -y install nginx                 77.8MB
470671670cac        4 days ago          /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>           4 days ago          /bin/sh -c #(nop)  LABEL org.label-schema.sc…   0B
<missing>           7 days ago          /bin/sh -c #(nop) ADD file:aa54047c80ba30064…   237MB
```



`FROM centos`

`RUN yum install nginx`

`RUN yum install php`

382 MB Total

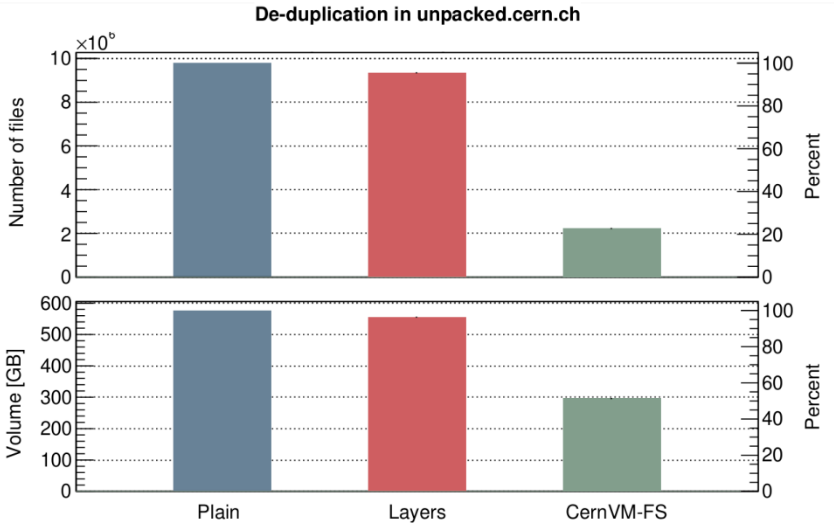
"[…] but only 6.4% of that data is read."

T. Harter, et al., "*Slacker: Fast Distribution with Lazy Docker Containers*" Usenix 2016

docker build –t myimage .

1.1 CVMFS for Containers – Efficiency

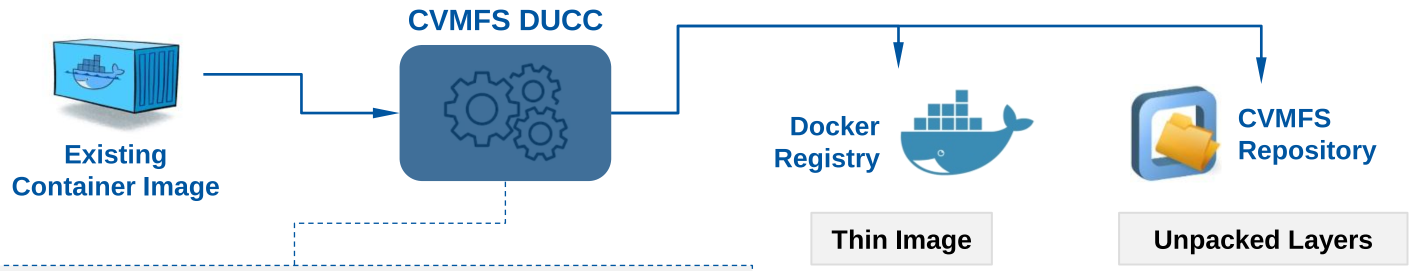
- **De-duplication** on ingestion
  - Deduplication with file-level granularity is more efficient than per-layer
  - Duplication occurs more often for smaller files



- **On-demand caching** on clients
  - No need to pull and extract images locally
  - Files are fetched from CVMFS when required
  - Smaller cache on client nodes
  - CVMFS self-manages local cache
    Traditional container images must be manually evicted

1.1 CVMFS for Containers – Server Ingestion

- DUCC: Daemon that Unpacks Container images into CVMFS
  - Downloads and unpacks existing container images
  - Publishes the obtained flat root file system into a CVMFS repository
  - Generates the *Thin Image* and pushes it to a Docker registry



```
cvmfs_repo: unpacked.cern.ch
output_format:
'$(scheme)://registry.gitlab.cern.ch/thin/$(image)'
input:
    - 'https://registry.hub.docker.com/library/fedora:latest'
    - 'https://registry.hub.docker.com/library/debian:stable'
```

1.1 CVMFS for Containers – Runtimes Integration

- CVMFS supports several container runtimes
  - **Flat runtime**: Starts container from unpacked root file system
  - **Layer runtime**: Constructs root file system from several directories

Runtime	Type	CVMFS Support
Singularity	Flat (+ Layers)	Native
runc	Flat (+ Layers)	Native
Docker	Layers	*Graph Driver* Plugin
containerd / k8s	Layers	Prototype
podman	Layers (+ Flat)	Prototype

2

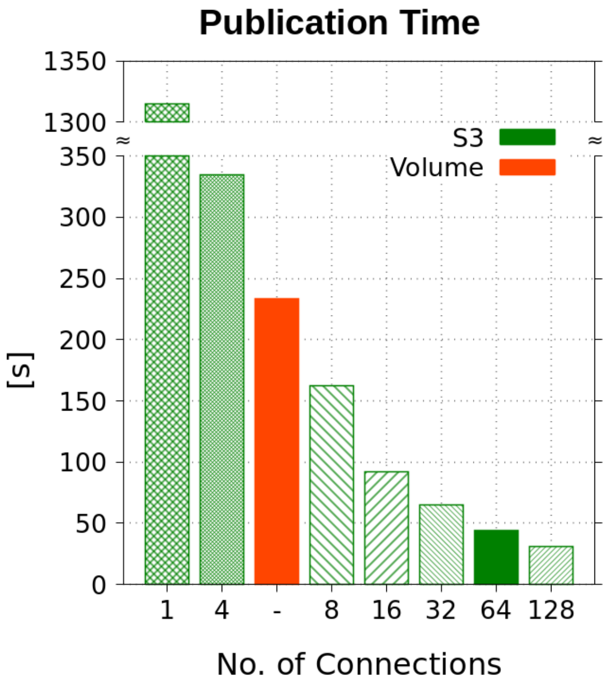
Infrastructure Improvements

1. S3 as Stratum 0 Storage
2. Dedicated Caches for Content Delivery

2.1 S3 as Stratum 0 Storage

- s3.cern.ch: Single-region RADOS Gateway cluster
  - ➢ Load-balanced across 16 VMs with Traefik/RGWs
  - ➢ Dedicated RGWs for CVMFS (and other use cases)

- Performance advantages
  - ➢ S3 with parallel uploads outperforms volume storage
  - ➢ Publication on S3 is **5x faster**
  - ➢ Publication time benchmarking
    - ✓ Sample workload: 250k files, 4 kB each
    - ✓ Files are organized in 250 folders
    - ✓ Time is full publication chain through cvmfs_server



- Operational advantages
  - ➢ Online quota management and extension
  - ➢ Easier failover of Stratum 0 to another server
  - ➢ Redundant and scalable HTTP access

2.1 S3 as Stratum 0 Storage

- S3 is the default storage for Stratum 0s since Q4 2018
  - 15 repositories created since then

- Ongoing migration campaign of existing repositories to S3
  - Many Stratum 0s running SLC6 (EOL 11/2020) migrated to CC7 + S3
  - 35 (out of 42) migrated during Q2 and Q3 2020
  - 1 B objects (80% of total), 46.32 TB (66% of total)
  - Critical repositories from major LHC experiments (atlas, lhcb, alice, …)

- Plan is to finalize migrations by the end of 2020
  - 7 repositories remaining, 5 planned for migration
  - Remove support for volumes to ease operations

2.2 Dedicated Caches

- 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

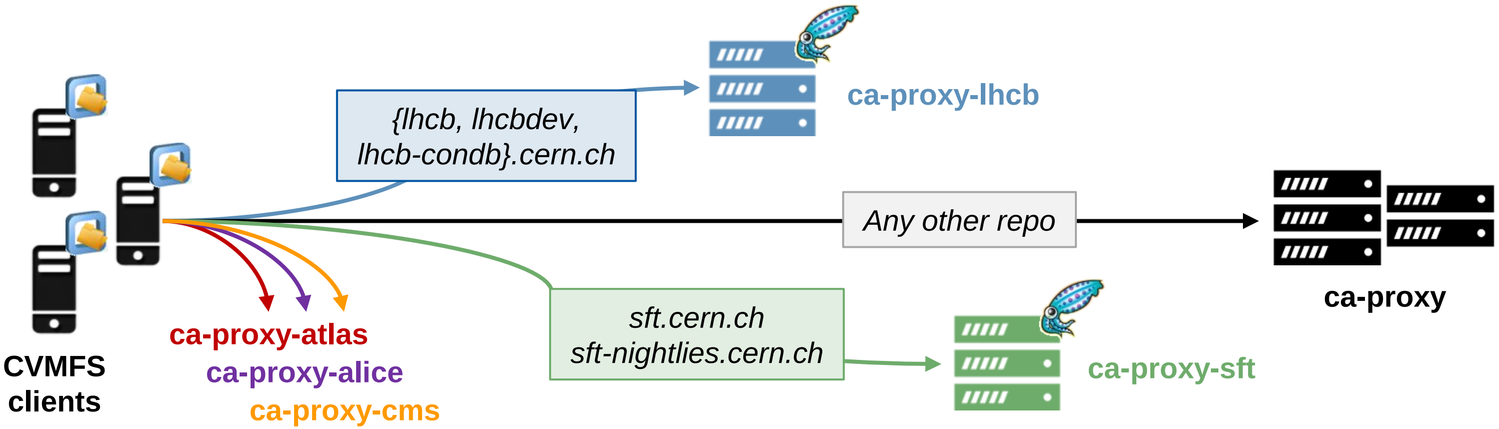
- Problem 1: Caches get inefficient (requests/traffic hit rates decrease)
  - Cache do not coordinate / peer. They all tend to cache the same items
  - Size of the repositories constantly increases, size of the caches does not

- 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

2.2 Dedicated Caches

- Goal: Reduce interference across repositories and improve cache efficiency

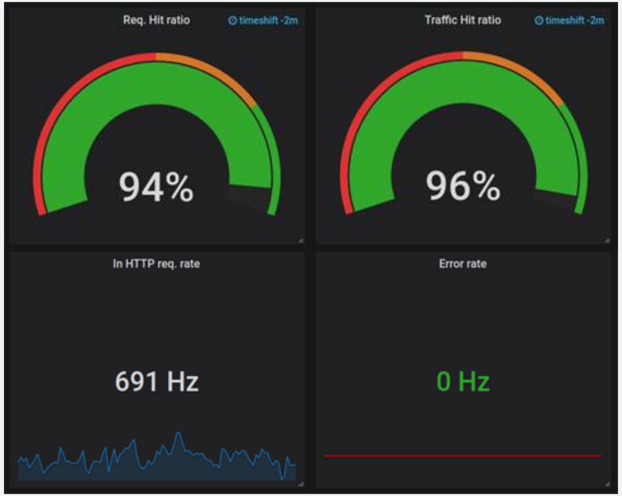
- Result: Dedicated caches for groups of repositories
  - 5 sub-pools of caches for 4 main LHC experiments (ca-proxy-alice, ca-proxy-atlas, …) + 1 for SFT
  - Several CNAMEs (e.g., ca-proxy-compass, ca-proxy-ams, …) to steer traffic in case they cause overloads
  - 1 pool of general caches remains for all other repos (ca-proxy.cern.ch)

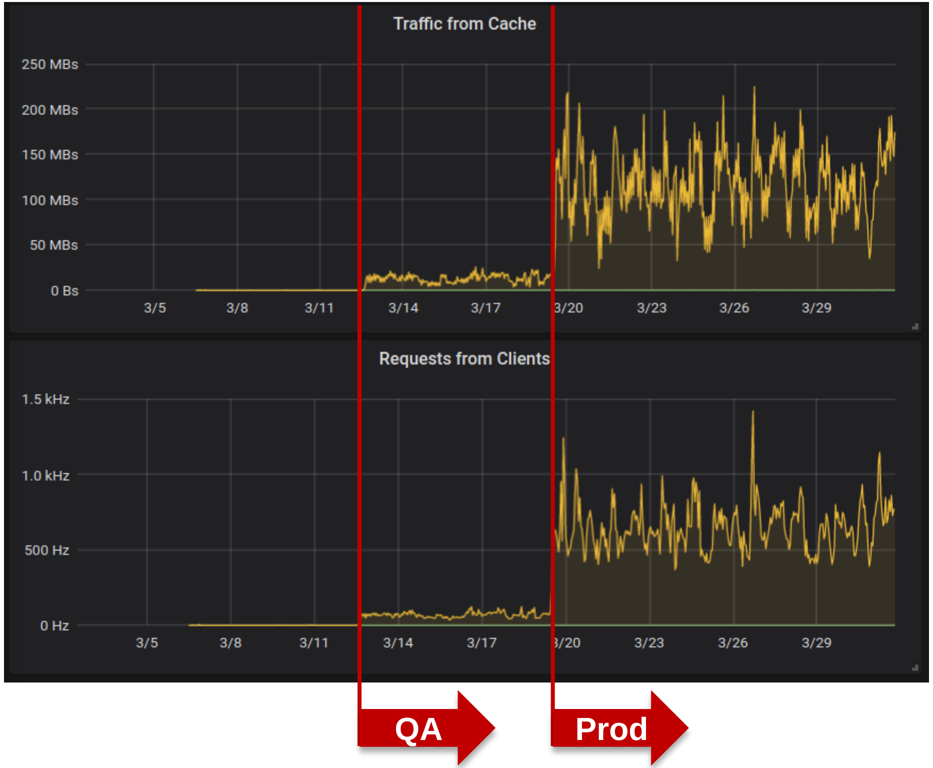


2.2 Dedicated Caches

- Example for LHCb repositories
  - **ca-proxy-lhcb.cern.ch**

Statistics over last 30 days





3

Closing Remarks

Conclusions

- CVMFS is a core service for software distribution at scale
  - At CERN and for the WLCG
  - Major experiments heavily relying on it
  - Ubiquitous client empowering diverse use cases

- Evolving with new capabilities and components
  - Ingestion and distribution of container layers

- Improvements in the infrastructure
  - Migration to S3 makes publications faster
  - Dedicated caches for more reliable distribution to clients

Thank you!

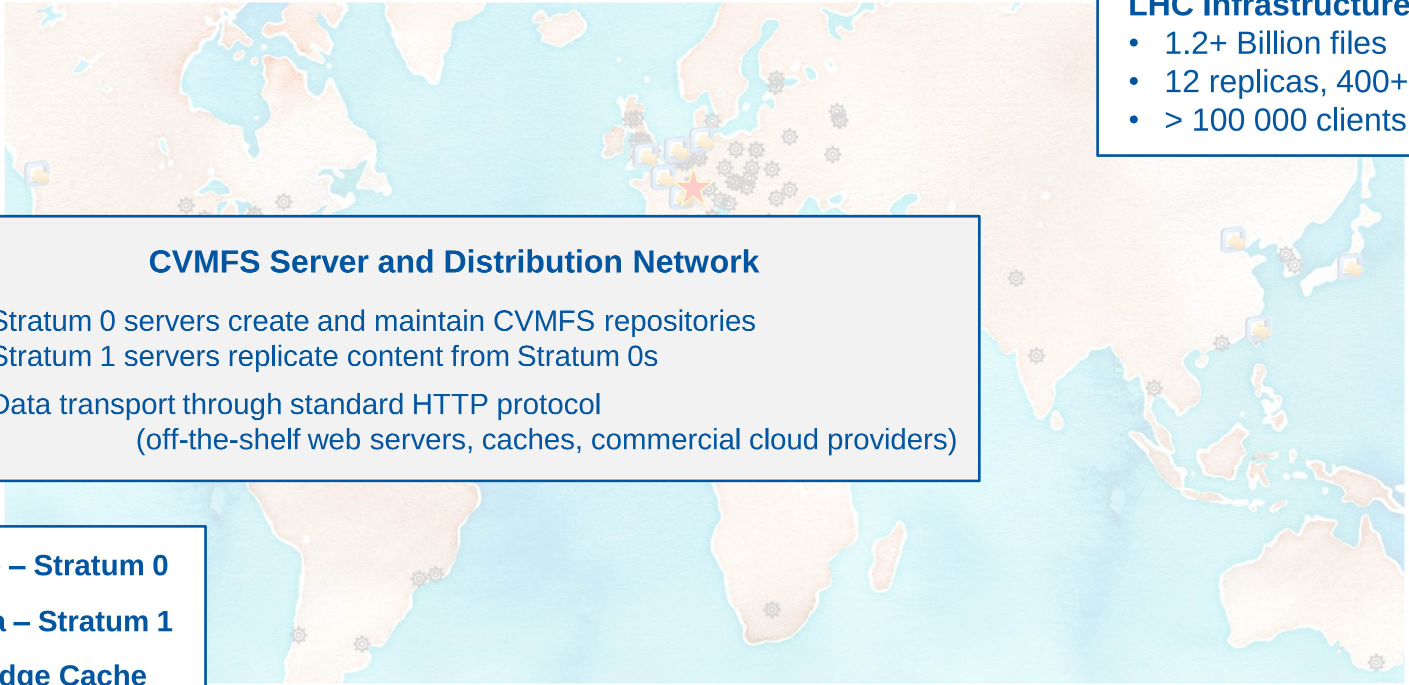
Questions?

Enrico Bocchi

enrico.bocchi@cern.ch

Backup

CVMFS in a Nutshell



LHC Infrastructure
- 1.2+ Billion files
- 12 replicas, 400+ caches
- > 100 000 clients

CVMFS Server and Distribution Network

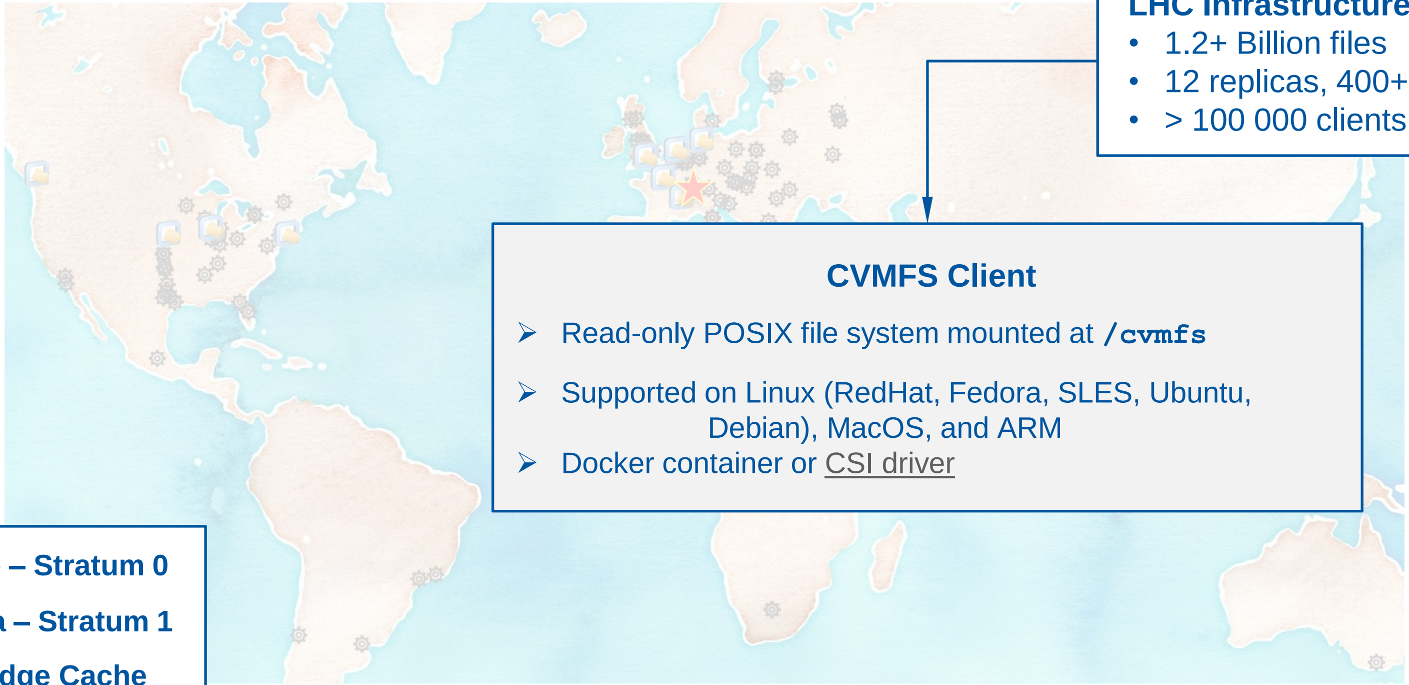
- Stratum 0 servers create and maintain CVMFS repositories
- Stratum 1 servers replicate content from Stratum 0s
- Data transport through standard HTTP protocol
  (off-the-shelf web servers, caches, commercial cloud providers)

★ **Source – Stratum 0**

▣ **Replica – Stratum 1**

⚙ **Site / Edge Cache**

CVMFS in a Nutshell



LHC Infrastructure

- 1.2+ Billion files
- 12 replicas, 400+ caches
- > 100 000 clients

CVMFS Client

- Read-only POSIX file system mounted at `/cvmfs`
- Supported on Linux (RedHat, Fedora, SLES, Ubuntu, Debian), MacOS, and ARM
- Docker container or CSI driver

★ **Source – Stratum 0**

▣ **Replica – Stratum 1**

⚙ **Site / Edge Cache**

S3 at CERN

- Single region RADOS Gateway cluster
  - 5000+ users, 2.1 PB raw capacity
  - 4+2 erasure coding for data, 3x replication for bucket indexes

  - s3.cern.ch load-balanced across 16 VMs with Traefik / RGWs
    - ✓ 5x general-purpose RGWs
    - ✓ 11x dedicated RGWs for specific use cases (e.g., 2x CVMFS, 3x GitLab, …)
    - ✓ Traefik as ingress to s3.cern.ch, routes traffic to dedicated RGWs

- Cluster upgraded BlueStore + bucket indexes on SSD (Q1 2019)
  - BlueStore's RocksDB on SSDs outperforms old FileStore's LevelDB on HDDs

  - Massive metadata performance increase
  - Metrics before were ~2kHz each!

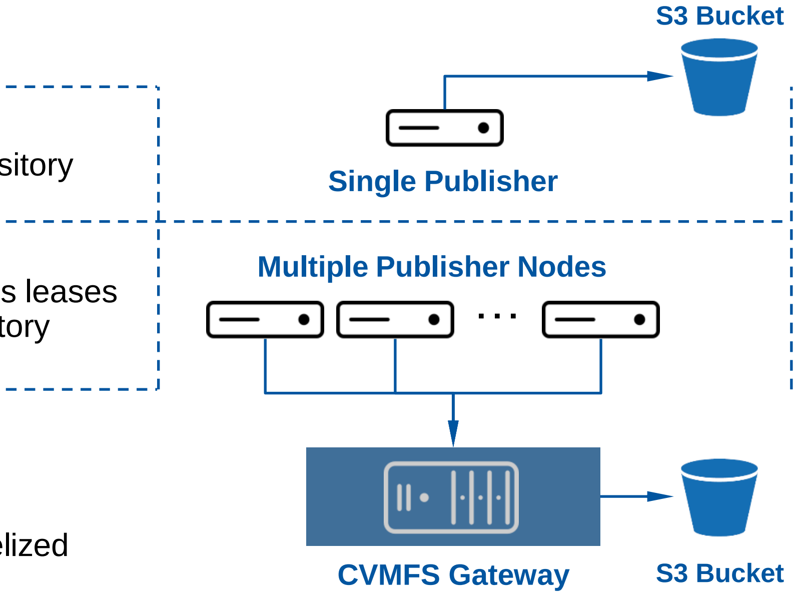
Metric	Rate
PUT (new)	83kHz ± 4kHz
HEAD (not found)	63kHz ± 2kHz
DELETE	198kHz ± 15kHz

CVMFS Gateway

- Stateful component allowing for concurrent publications
  - Issues time-limited leases for specific sub-paths
  - Has exclusive access to repository storage

- Typical CVMFS setup
  - One publisher has global lock when writing to the repository

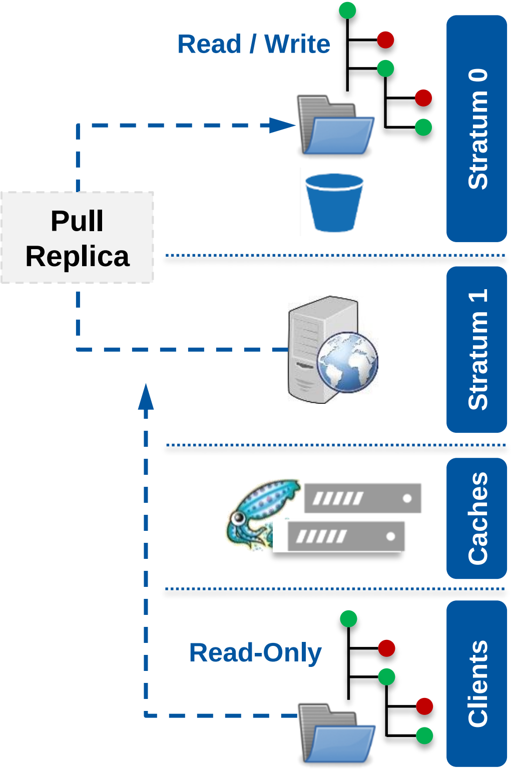
- CVMFS setup with Gateway
  - One Gateway regulates access to storage and provides leases
  - Multiple publishers publish concurrently into the repository



- Relevant for Integration Builds repositories
  - Reduced time to publish all nightly builds
  - Benefits publication pipelines that can be easily parallelized

Pass-Through Repositories

- Typically, clients read from Stratum 1 (through caches):
  - Protect the Stratum 0 server from client traffic
  - Stratum 1 replicates content periodically from Stratum 0
  - A (very small) replication delay exists between Stratum 0 and 1
  - Stratum 1 might lag behind when garbage collecting

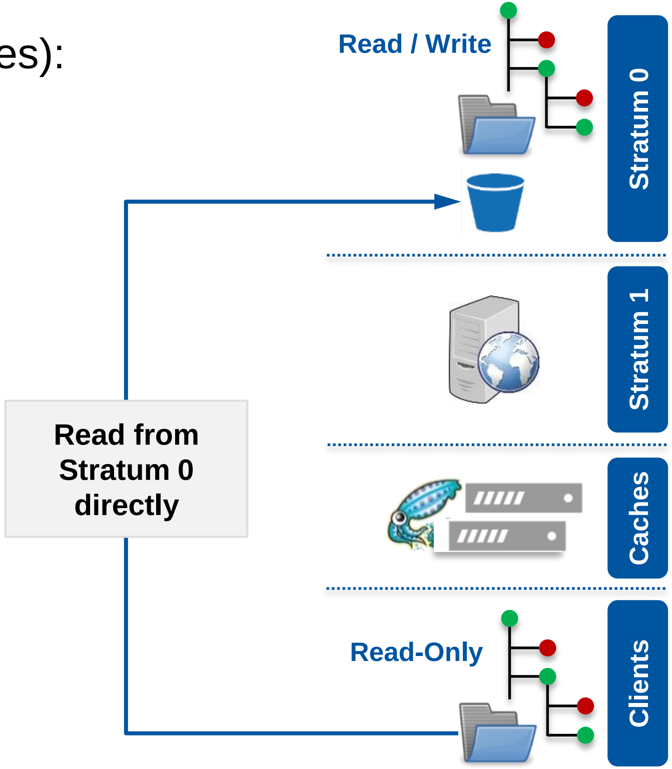


Pass-Through Repositories

- Typically, clients read from Stratum 1 (through caches):
  - Protect the Stratum 0 server from client traffic
  - Stratum 1 replicates content periodically from Stratum 0
  - 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 the Stratum 0:
  - No need to replicate to Stratum 1 – No replication delay
  - Garbage collection is not blocking for reads

- Relevant for cms-ib.cern.ch:
  - Other pipelines depend on what is published on CVMFS
  - Would like to have changes in CVMFS immediately visible



Read / Write
Stratum 0
Stratum 1
Caches
Read-Only
Clients
Read from Stratum 0 directly