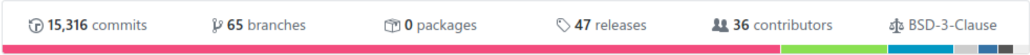
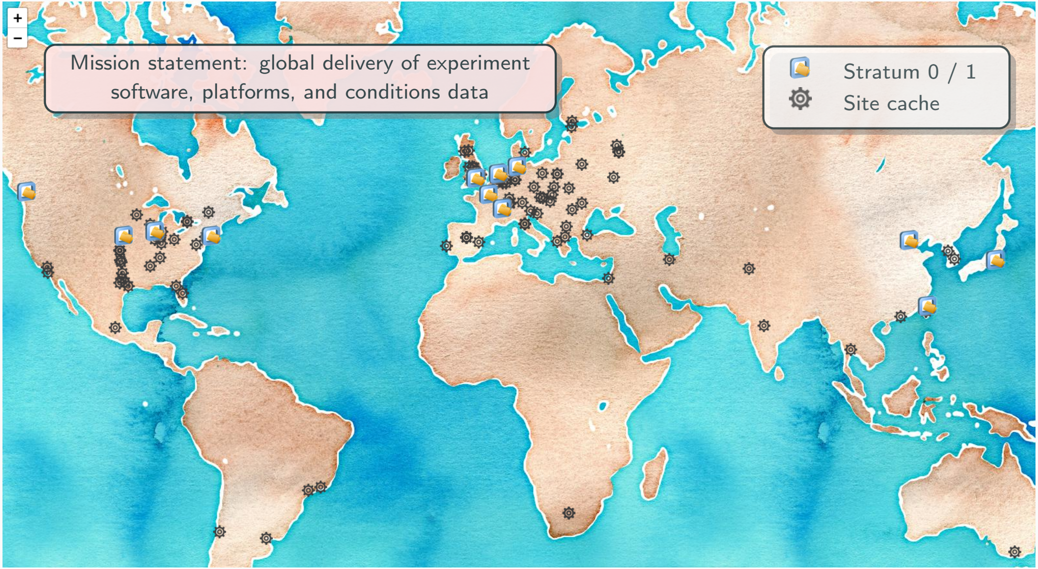
CernVM-FS Status and Container Integration Update

J Blomer for the CernVM Team

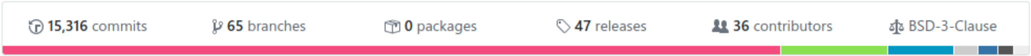
Pre-GDB on Software Deployment
5 May 2020



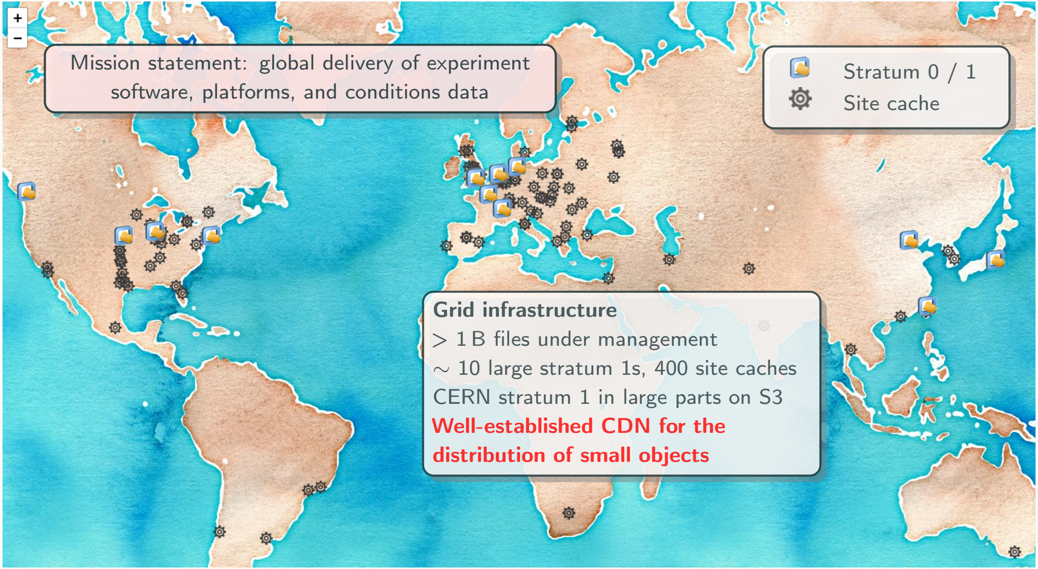
15,316 commits | 65 branches | 0 packages | 47 releases | 36 contributors | BSD-3-Clause



Mission statement: global delivery of experiment software, platforms, and conditions data



15,316 commits | 65 branches | 0 packages | 47 releases | 36 contributors | BSD-3-Clause



Mission statement: global delivery of experiment software, platforms, and conditions data

Stratum 0 / 1
⚙ Site cache

Grid infrastructure

> 1 B files under management

∼ 10 large stratum 1s, 400 site caches

CERN stratum 1 in large parts on S3

Well-established CDN for the distribution of small objects

Principal Content Types

❶ Production Software

Example: */cvmfs/atlas.cern.ch*

- ✓ Most mature use case
- ⛅ Containerizing publish workflows

❷ Integration Builds

Example: */cvmfs/lhcbdev.cern.ch*

- ✓ High churn, requires regular garbage collection
- ⛅ Improving pipelines (build – deploy – test)

❸ Unpacked Container Images

Example: */cvmfs/singularity.opensciencegrid.org*

- ✓ Works out of the box with Singularity
- ✓ CernVM-FS plugin for Docker
- ⛅ Integration with podman, containerd / k8s

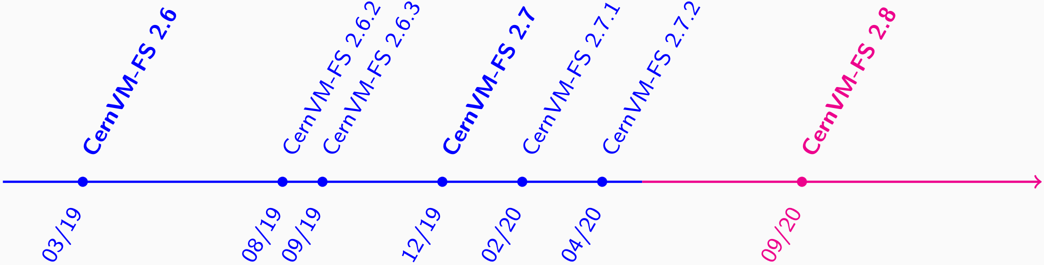
❹ Auxiliary data sets

Example: */cvmfs/alice-ocdb.cern.ch*

- ✓ Benefits from internal versioning
- ● Depending on volume requires more planning for the CDN components

⛅ Current focus of developments

Release Plan



Current development lines focus on containerizing the publishing process

CernVM-FS integration with container runtimes (singularity, containerd, podman, . . .)
takes place in parallel to releases

Containers and CernVM-FS

Containers and CernVM-FS

❶ CernVM-FS in containers

- Bind mount:

```
docker run -v /cvmfs:/cvmfs:shared ...
singularity exec -B /cvmfs ...
```

- CSI driver ▸ Github repository
  "behind the scenes" bind mount, integrates with kubernetes (maintained by IT)

- **New: unprivileged mounting inside container**
  Attractive option on opportunistic resources with the advent of user-level fuse mounts (EL >7.8); challenge on sharing the cache among containers
  → see `cvmfsexec` talk later

❷ Container images in CernVM-FS

Unpacked images on /cvmfs in order to benefit from de-duplication and on-demand caching

Requires:

1. Container image conversion
   → see DUCC and unpacked.cern.ch talk later
2. Storage plug-in required for layer based container runtimes

Reminder:

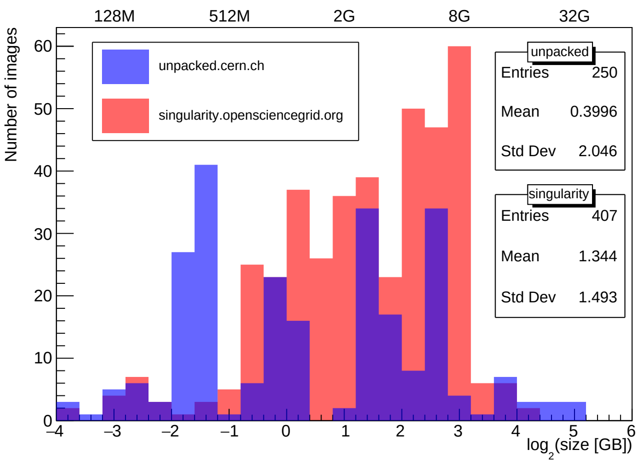
- "*Flat runtime*": starts container from unpacked root file system (e. g. singularity, runc)
- "*Layer runtime*": constructs root file system with Overlay-FS from several directories (e. g. docker, containerd)

Use cases ❶ and ❷ can be combined

Container Image Sizes

Distribution of container images sizes in
`/cvmfs/unpacked.cern.ch` and `/cvmfs/singularity.opensciencegrid.org`

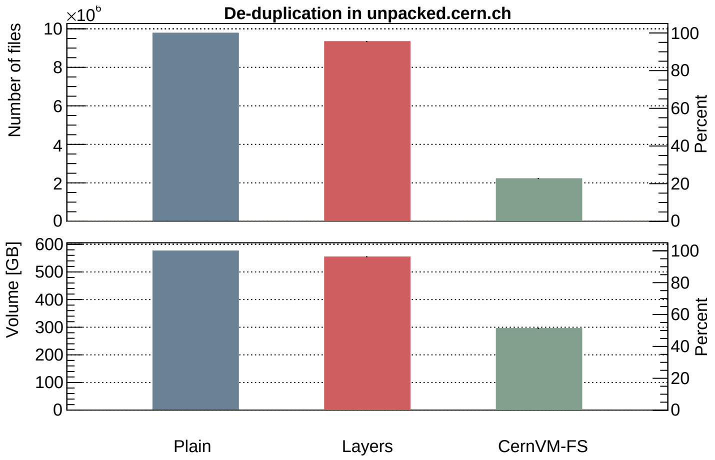
- **Likely to overflow the worker node scratch space with multi-gigabyte images**
- Interesting follow-up: distribution by image category (base, user, . . .) and creation date



De-duplication

Comparison of de-duplication efficiency between layers and file-based storage (CernVM-FS)

- **De-duplication works properly only on file-level granularity**
- Duplication occurs more often for smaller files
- Interesting follow-up: de-duplication in worker node caches



Worker Node Software Space

First observations from CERN lxbatch farm:

- Looked at 3 different images (ATLAS and CMS base images)
- Found on > 250 worker nodes
- For each image:
  - 2 % to 9 % of the **image volume** in the worker node cache
  - Site-wide: ∼ 15 % of the volume cached
- Not yet including de-duplication effects in the worker node cache

→ **×10 to ×50 higher image distribution efficiency from /cvmfs hosted images compared to** `docker pull ...`

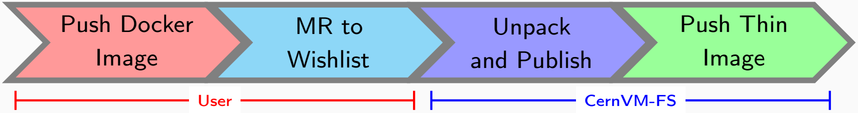
Container Runtime Integration

Runtime	**Type**	**CernVM-FS Support**
Singularity	flat (+ layers)	**native**
runc	flat (+ layers)	**native**
docker	layers	***“graph driver”* image storage plugin**
containerd / k8s	layers	**prototype**
podman	layers (+ flat)	**GSoC project in collaboration with podman engineers**

Currently improving documentation, examples, integration tests for different deployment options
→ `https://cvmfs.readthedocs.io/en/latest/cpt-containers.html`

→ See Simone’s talk

DUCC Container Publishing Service: Workflow



Wishlist https://gitlab.cern.ch/unpacked/sync

```yaml
version: 1
user: cvmfsunpacker
cvmfs_repo: 'unpacked.cern.ch'
output_format: >
  https://gitlab-registry.cern.ch/unpacked/sync/$(image)
input:
    - 'https://registry.hub.docker.com/library/fedora:latest'
    - 'https://registry.hub.docker.com/library/debian:stable'
    - 'https://registry.hub.docker.com/library/centos:*'
```

Multiple wishlists possible, e. g. experiment specific

/cvmfs/unpacked.cern.ch

```
# Singularity
/registry.hub.docker.com/fedora:latest -> \
  /cvmfs/unpacked.cern.ch/.flat/d0/d0932...
# Docker with thin image
/.layers/f0/1af7...
```

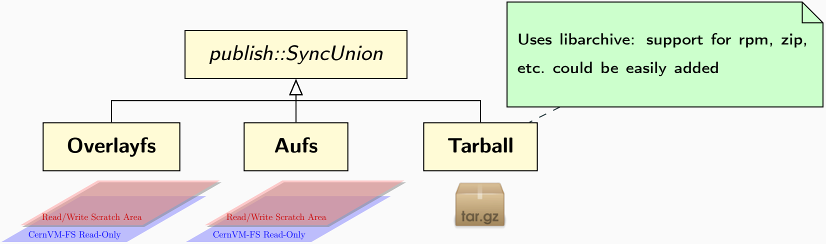
→ see Simone's talk for current developments and possible future alternative conversion tools

Enabling Feature for Container Publishing: Tarball Ingestion

Direct path for the common pattern of publishing tarball contents

```
$ cvmfs_server transaction
$ tar -xf ubuntu.tar.gz
$ cvmfs_server publish
```

```
$ cat ubuntu.tar.gz | \
    cvmfs_server ingest -t -
```



Uses libarchive: support for rpm, zip, etc. could be easily added

Performance Example

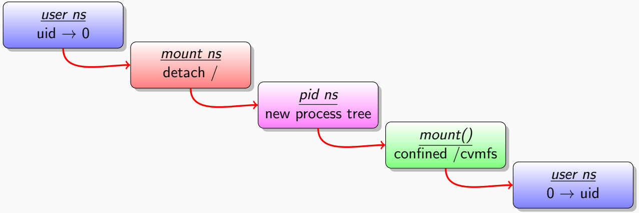
Ubuntu 18.04 container – 4 GB in 250 k files: **56 s untar + 1 min publish vs. 74s ingest**

CernVM-FS Access on Foreign Resources (Resources not Controlled by HEP)

Opportunistic Resources: Unprivileged Mounting

Technical foundations

- User namespaces completing container support
- As of Linux kernel version 4.18 (EL8, but also EL 7.8), **fuse mounts are unprivileged in user name spaces**
- Overlay-FS implementation available as a fuse module



user ns
uid → 0

mount ns
detach /

pid ns
new process tree

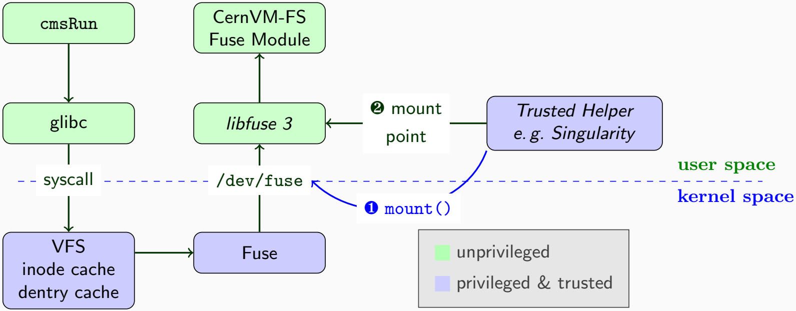
mount()
confined /cvmfs

user ns
0 → uid

→ see Dave's talk on `cvmfsexec`

HPC Singularity Integration

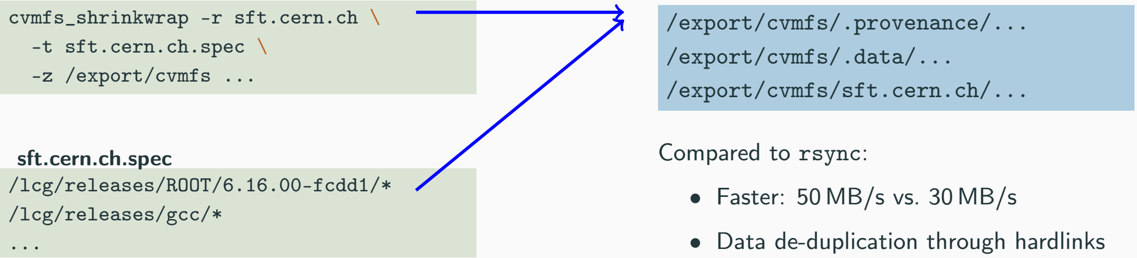
- With the new Fuse3 libraries, the task of mounting /dev/fuse can be handed to a trusted, external helper.
- Fuse3 libraries have been backported to EL6 and EL7 platforms.
- Gives access to /cvmfs in containers started by singularity



Pre-mounting is implemented in **Singularity 3.4** and **CernVM-FS 2.7**

HPC “Fat Containers”: Shrinkwrap

Export bulky /cvmfs subtrees into “fat containers”.
Used by CMS for US HPCs, also used by IT/HEPiX benchmark working group.



```
cvmfs_shrinkwrap -r sft.cern.ch \
  -t sft.cern.ch.spec \
  -z /export/cvmfs ...
```

sft.cern.ch.spec

```
/lcg/releases/ROOT/6.16.00-fcdd1/*
/lcg/releases/gcc/*
...
```

```
/export/cvmfs/.provenance/...
/export/cvmfs/.data/...
/export/cvmfs/sft.cern.ch/...
```

Compared to rsync:

- Faster: 50 MB/s vs. 30 MB/s
- Data de-duplication through hardlinks
- Efficient synchronization and GC
- Aware of CernVM-FS specifics

Shrinkwrapping is a rather heavy-weight process, worthwhile only for special cases

Summary

- CernVM-FS provides a central software deployment hub
- Software bookkeeping and preservation built-in
- Automatic worker node cache management
- Well-established, secure and efficient content distribution:
  ×10 to ×50 improvement compared to plain container distribution

- **Linux is solving the problem of unprivileged CernVM-FS fuse mounts**
- **Ongoing work on improving container image conversion to close the gap between images in the registry and unpacked images in /cvmfs**
- **On track with container runtime plugins to support cvmfs based image distribution**