CernVM-FS Overview and Roadmap

Jakob Blomer, CERN

6th EasyBuild User Meeting

25 January 2021

Agenda

CernVM-FS LHC Deployment

CernVM-FS LHC Deployment

Software Delivery by CernVM-FS

❶ CernVM-FS provides uniform, consistent, and versioned POSIX file system access to `/cvmfs`

```
$ ls /cvmfs/cms.cern.ch
slc7_amd64_gcc700  slc7_ppc64le_gcc530  slc7_aarch64_gcc700  slc6_mic_gcc481
...
```

on **grids**, **clouds**, **supercomputers** and **end user laptops**

read

---

publish

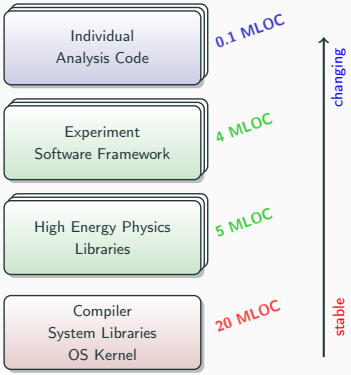
❷ Populate and propagate new and updated content

- A few "software librarians" can publish into /cvmfs
- All content in /cvmfs is cryptographically signed
- Transactional writes as in `git commit/push`

The Software Distribution Challenge

The Anatomy of a Scientific Software Stack


```
$ cmsRun DiPhoton_Analysis.py
```



Key Figures for LHC Experiments

- Hundreds of (novice) developers
- $>$ 100 000 files per release
- 1 TB / day of nightly builds
- $\sim$100 000 machines world-wide
- Daily production releases, remain available "eternally"

Common Tools for Sharing Data Center Applications

Applications are **bundled** (container, package, . . .) and **installed** where needed.

Bundles structure the build process but are inefficient for synchronized, large-scale rollout

Common Tools for Sharing Data Center Applications

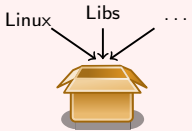
Applications are **bundled** (container, package, . . .) and **installed** where needed.

Bundles structure the build process but are inefficient for synchronized, large-scale rollout

Example: R bundled in Docker

Example: R in Docker

```
$ docker pull r-base
```
⟶ 1 GB image
```
$ docker run -it r-base
$ ... (fitting tutorial)
```
⟶ only 30 MB used



Ideally: Containers for isolation and orchestration, but not for distribution

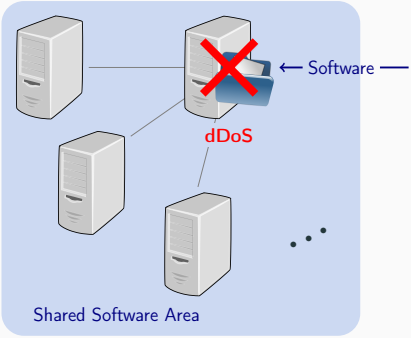
Shared Software Area on General Purpose DFS

Working Set

- ≈2 % to 10 % of all available files of a software release are requested at runtime
- Median of file sizes: $< 4\,\text{kB}$

Flash Crowd Effect

- $\mathcal{O}$(MHz) meta data request rate
- $\mathcal{O}$(kHz) file open rate



Software vs. Data

Software	Data
POSIX interface	put, get, seek, streaming
File dependencies	Independent files
O(kB) per file	O(GB) per file
Whole files	File chunks
Absolute paths	Relocatable
WORM ("write-once-read-many")	
Billions of files	
Versioned	

Software is massive not in volume but in number of objects and meta-data rates

CernVM-FS: A Purpose-Built Software File System

Principal Application Areas

❶ Production Software

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

- Relatively stable repositories
- Often used for large-scale data processing jobs

❷ Integration Builds

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

- High churn, usually accessed from developer machines

❸ Unpacked Container Images

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

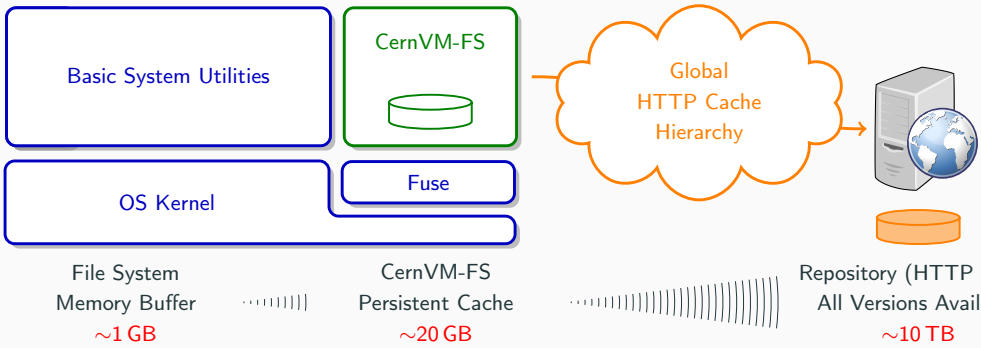
- Enables large scale container deployment with Singularity (and other container runtimes)

❹ Auxiliary data sets

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

- Somewhat larger data set than software but similar access pattern

Reading

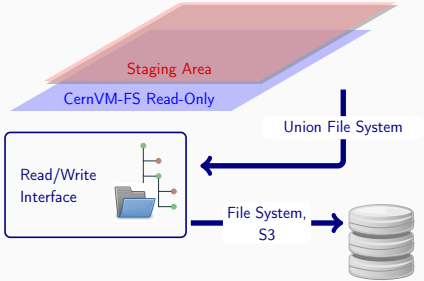


- Fuse based, independent mount points, e. g. /cvmfs/atlas.cern.ch
- High cache effiency because entire cluster likely to use same software

Client Availability

- *A* Platforms:
  - EL 7–8 AMD64
  - Ubuntu 16.04, 18.04, 20.04 AMD64
- *B* Platforms
  - macOS 10.15, 11
  - SLES 11 – 12
  - Fedora, latest two versions
  - Debian 8–10
  - EL7 AArch64
  - IA32 architecture
  - Linux on Windows via the WSL-2 subsystem
- Experimental: POWER, Raspberry Pi, RISC-V

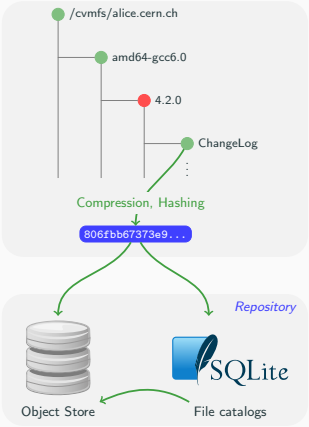
Publishing (Writing)



Publishing new content

```
[ ~ ]# cvmfs_server transaction containers.cern.ch
[ ~ ]# cd /cvmfs/containers.cvmfs.io && tar xvf ubuntu1610.tar.gz
[ ~ ]# cvmfs_server publish containers.cern.ch
```

Use of Content-Addressable Storage



⊕ Immutable files, trivial to check for corruption, versioning, efficient replication

⊖ compute-intensive, garbage collection required

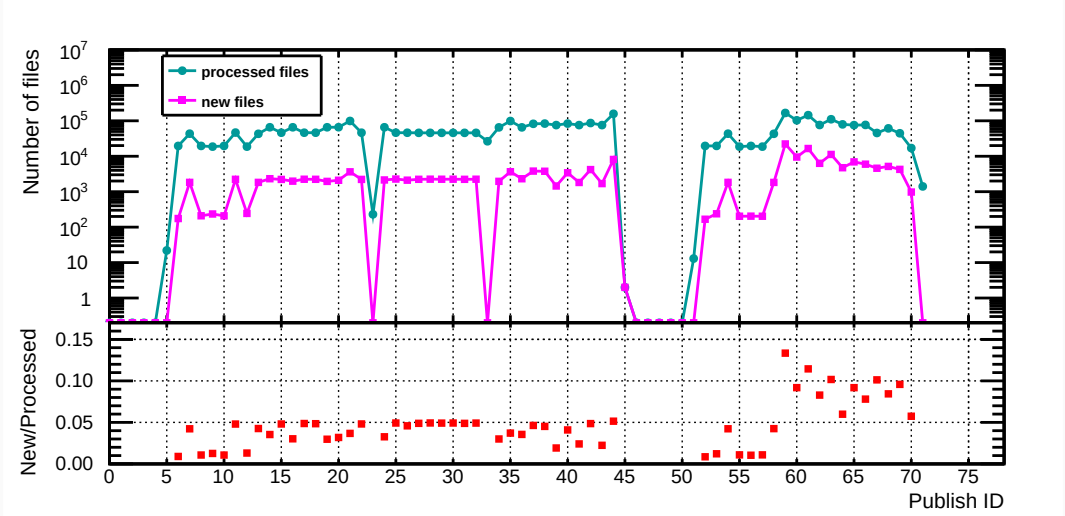
Object Store

- Compressed files and chunks
- De-duplicated

File Catalog

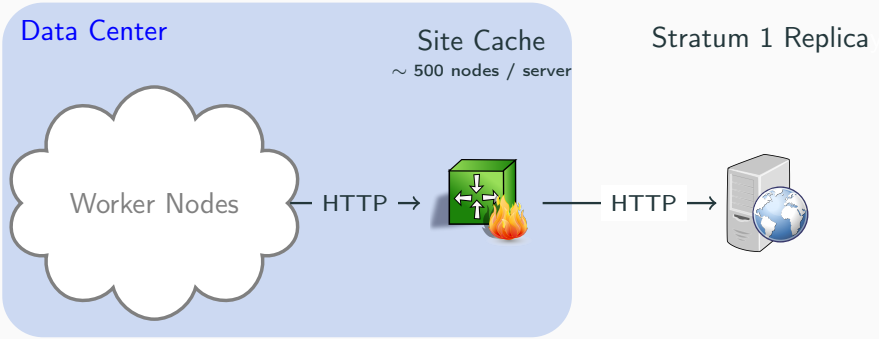
- Directory structure, symlinks
- Content hashes of regular files
- Large files: chunked with rolling checksum
- Digitally signed
- Time to live
- Partitioned / Merkle hashes (possibility of sub catalogs)

Repository Statistics: File De-Duplication



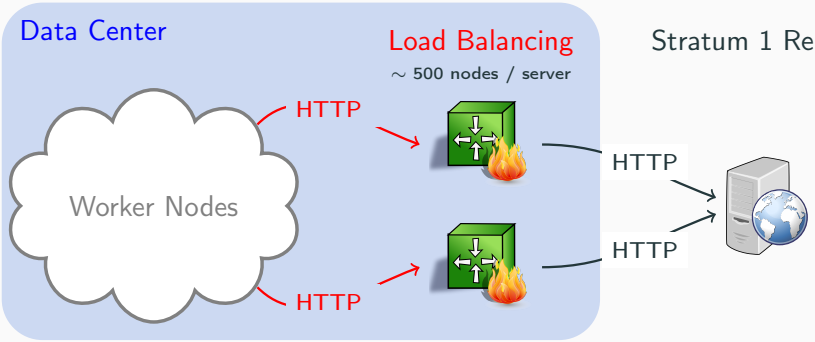
High-Availability by Horizontal Scaling

Server side: stateless services



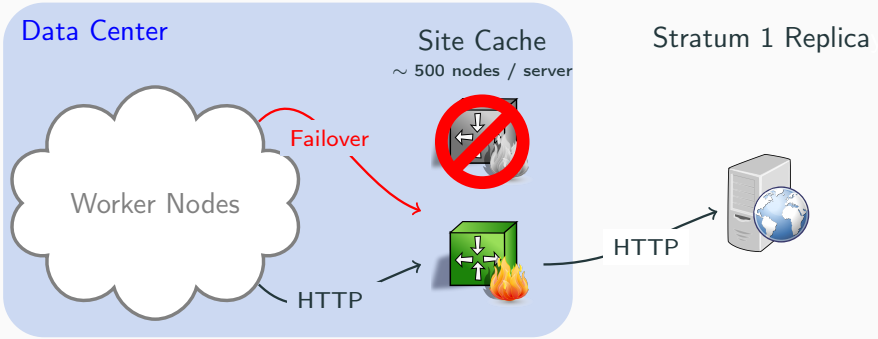
High-Availability by Horizontal Scaling

Server side: stateless services



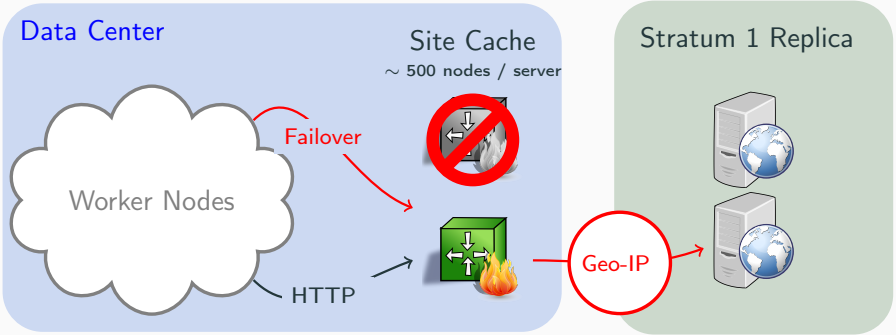
High-Availability by Horizontal Scaling

Server side: stateless services



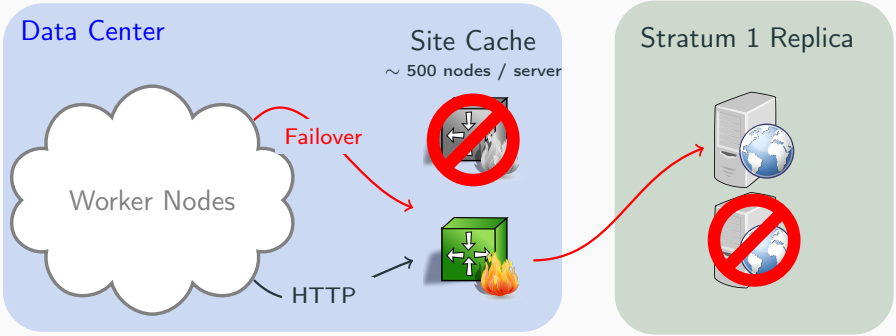
High-Availability by Horizontal Scaling

Server side: stateless services



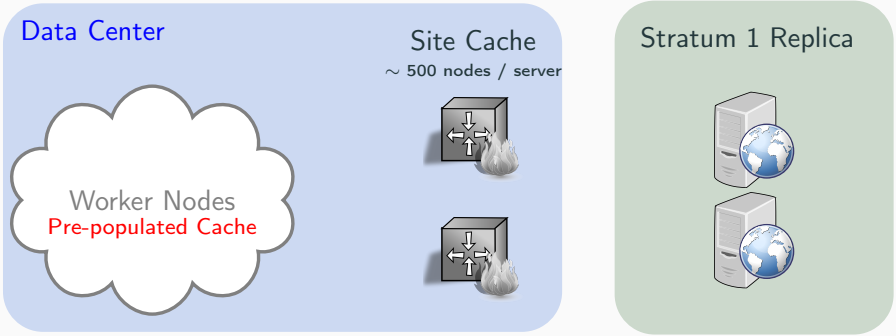
High-Availability by Horizontal Scaling

Server side: stateless services



High-Availability by Horizontal Scaling

Server side: stateless services



Containers and CernVM-FS

Containers and CernVM-FS: Overview

❶ Provide /cvmfs in a container

If /cvmfs is available on the host

- Bind mount from host to container

On opportunistic resources:

- **Unprivileged mounting inside container**
  Uses user-level fuse mounts (EL >7.8);
  challenge on sharing the cache among containers
- **Pre-mounted by singularity**
- **CernVM-FS "service container"**
  CernVM-FS client pre-packaged as a container

❷ Provide container from /cvmfs

Unpacked images on /cvmfs for scalable distribution

Requires:

1. Container image conversion:
   automated with the CernVM-FS DUCC service
2. Container runtime plug-in required for layered images

Reminder:

- "*Flat image*": starts container from unpacked root file system
- "*Layered image*": constructs root file system with Overlay-FS from several directories

Use cases ❶ and ❷ can be combined

Simple Case: CernVM-FS Available on the Host

```
$ docker run -v /cvmfs:/cvmfs:shared busybox ls /cvmfs/sft.cern.ch
README.md lcg
```

```
$ singularity exec -B /cvmfs docker://busybox ls /cvmfs/sft.cern.ch
README.md lcg
```

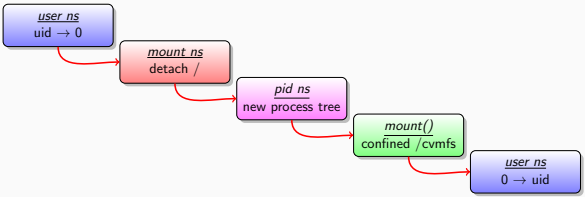
Important: use *shared* bind mount with docker so that that repositories can be mounted on demand from inside the container

Unprivileged Mounting with `cvmfsexec`

```
$ cvmfsexec grid.cern.ch atlas.cern.ch -- ls /cvmfs
atlas.cern.ch cvmfs-config.cern.ch grid.cern.ch
```

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



CernVM-FS Service Container

- CernVM-FS client in a minimal container
- Mounted /cvmfs inside container can be “leaked” to the outside host
- Alternative to system package based installation, e.g. on container-only operating systems
- Foundation of the kubernetes daemonset deployment ▸ sample deployment

```
$ docker run -d --rm \
    -e CVMFS_CLIENT_PROFILE=single -e CVMFS_REPOSITORIES=sft.cern.ch \
    --cap-add SYS_ADMIN --device /dev/fuse \
    --volume /cvmfs:/cvmfs:shared
    cvmfs/service
$ ls /cvmfs/
cvmfs-config.cern.ch sft.cern.ch
```

For HPCs: Pre-mounted by Singularity

- With the new Fuse3 libraries, mounting can be handed off to a trusted, external helper.
- Fuse3 libraries have been backported to EL6 and EL7 platforms.
- Gives access to /cvmfs in containers started by singularity (`singularity --fusemount`)
- **Required cvmfs client to be installed and prepared in the container**

```
$ CONFIGREPO=config-osg.opensciencegrid.org
$ mkdir -p $HOME/cvmfs_cache
$ singularity exec -S /var/run/cvmfs -B $HOME/cvmfs_cache:/var/lib/cvmfs \
  --fusemount "container:cvmfs2 $CONFIGREPO /cvmfs/$CONFIGREPO" \
  --fusemount "container:cvmfs2 sft.cern.ch /cvmfs/sft.cern.ch" \
  docker://davedykstra/cvmfs-fuse3 ls /cvmfs/sft.cern.ch
README.md lcg
```

CernVM-FS as a Container Hub

/cvmfs/unpacked.cern.ch

- > 700 images
- > 3TB
- > 50M files

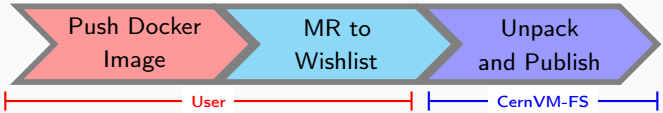
/cvmfs/singularity.opensciencegrid.org

- > 500 images
- > 2TB
- > 40M files

Images are readily available to run with singularity,
including **base operating systems**, **experiment software stacks**, **explorative tools (ML etc.)**,
user analyses, and special-purpose containers such as **folding@home**

```
[jblomer@lxplus.cern.ch]$ singularity exec \
  '/cvmfs/unpacked.cern.ch/registry.hub.docker.com/library/debian:stable' \
  cat /etc/issue
Debian GNU/Linux 10 \n \l
```

Container Conversion Service



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

```
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

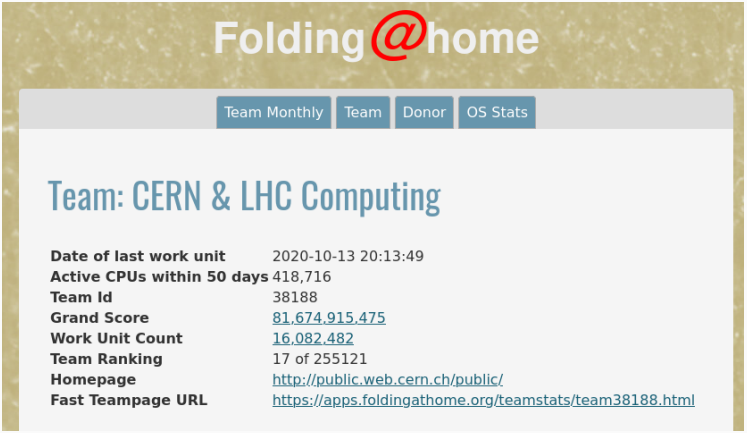
```
# Singularity
/registry.hub.docker.com/fedora:latest -> \
  /cvmfs/unpacked.cern.ch/.flat/d0/d0932...
# containerd, k8s, podman
/.layers/f0/1af7...
```

Ongoing work on direct registry integration, i. e. `docker push` triggers image conversion (ETA 2021)

Container Runtime Integration

Runtime	Type	CernVM-FS Support
Singularity	flat (+ layers)	**native**
podman	layers (+ flat)	**native**
docker	layers	***"graph driver"* image storage plugin**
containerd / k8s	layers	**pre-release** ▸ remote snapshotter

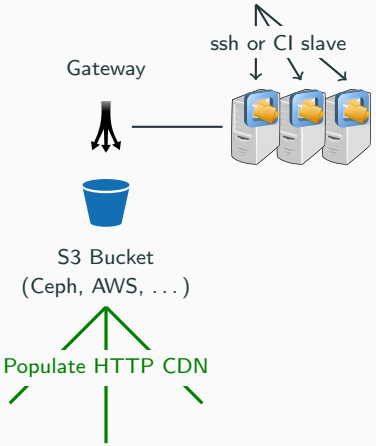
Example for Large-Scale Deployment: Folding@Home



Runs on the LHC computing grid off containers served from /cvmfs

Outlook

Distributed Publishing



Coordinating Multiple Publisher Nodes

- Concurrent publisher nodes access storage through gateway services
- Gateway services:
  - **API** for publishing
  - Issues **leases** for sub paths
  - Receives change sets as set of **signed object packs**

Ongoing work on stabilization and matching feature set

Ephemeral Publish Container

- A new command, `cvmfs_server enter`, creates a sub-shell with a writable /cvmfs
- Uses internally user namespaces and fuse-overlayfs
- Works unprivileged on any modern Linux that can mount the client
- Meant to be used on build nodes
- **Ongoing work to integrate with gateway publisher**

```
$ cvmfs_server enter hsf.cvmfs.io
...Opens a shell with write access to /cvmfs/hsf.cvmfs.io
$ cvmfs_server diff --workdir | ... | gzip > changes.tar.gz
...Back to read-only mode
```

Summary

Summary

- CernVM-FS: special-purpose virtual file system that provides a global shared software area for scientific collaborations
- Content-addressed storage and asynchronous writing (*publishing*) key to meta-data scalability
- Current areas of development:
  - Close integration with container engines
  - Scaling up number of writers

 https://cernvm.cern.ch/fs

 https://github.com/cvmfs/cvmfs