CERN's Virtual File System for Global-Scale Software Delivery

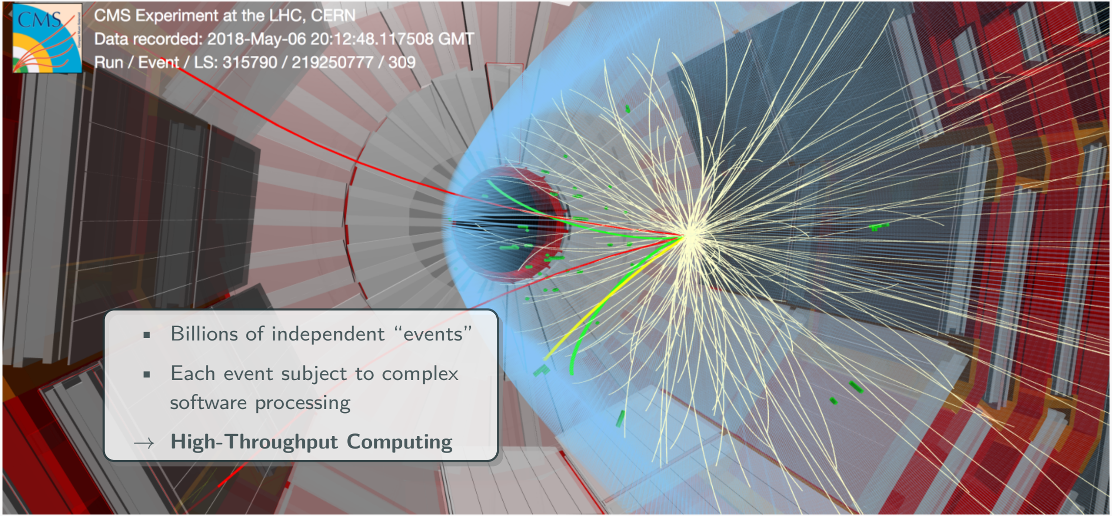
Jakob Blomer for the CernVM-FS team
CERN, EP-SFT

MSST 2019, Santa Clara University

Agenda

High Energy Physics Computing Model

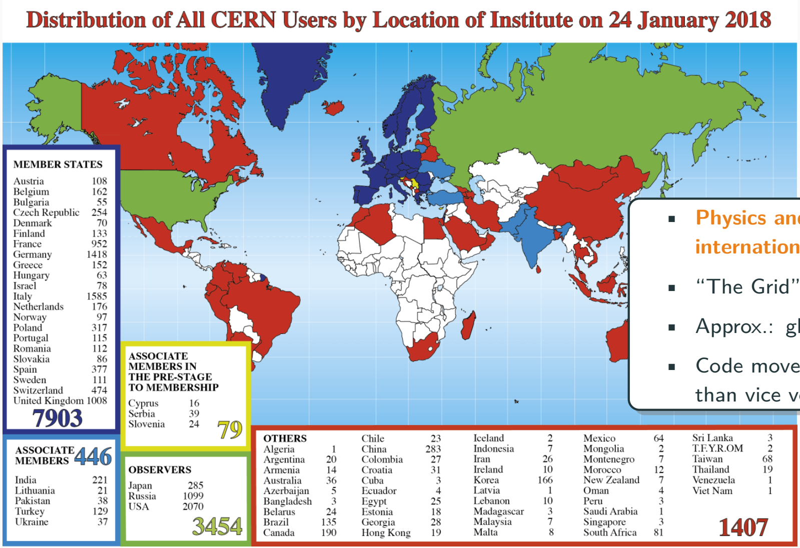
Measure & Analyze



- Billions of independent "events"
- Each event subject to complex software processing

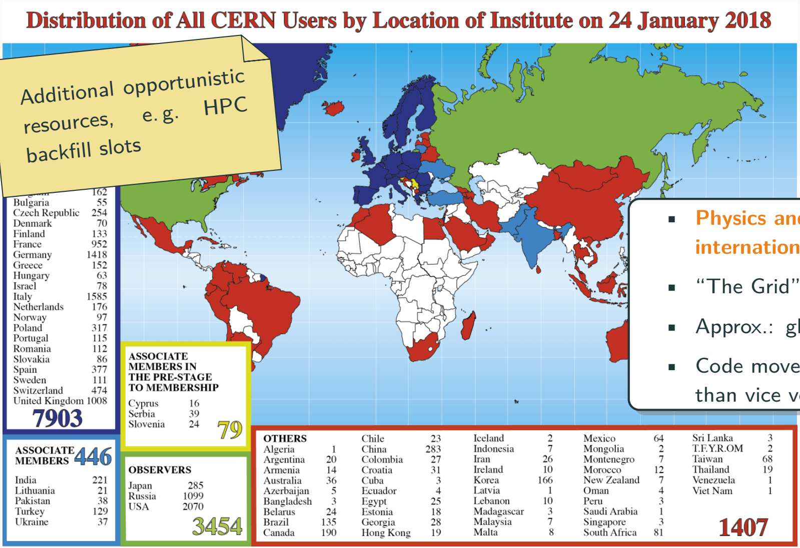
→ **High-Throughput Computing**

Federated Computing Model



- **Physics and computing: international collaborations**
- "The Grid": ∼160 data centers
- Approx.: global batch system
- Code moves to the data rather than vice versa

Federated Computing Model



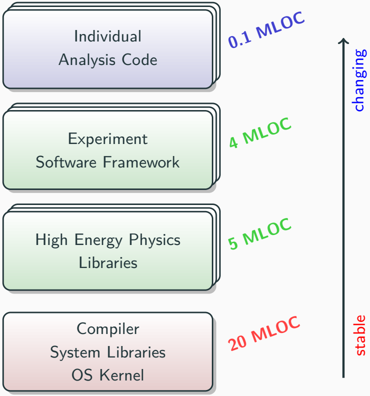
Additional opportunistic resources, e. g. HPC backfill slots

- **Physics and computing: international collaborations**
- “The Grid”: ∼160 data centers
- Approx.: global batch system
- Code moves to the data rather than vice versa

Software Distribution Challenge

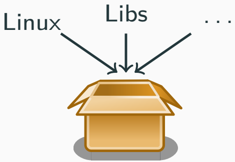
The Anatomy of a Scientific Software Stack

```
$ 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”



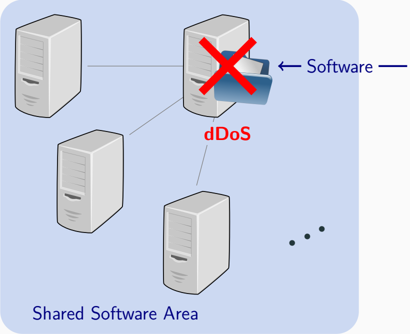
- Containers are easier to create than to role-out at scale
- Due to network congestion: long startup-times in large clusters
- Impractical image cache management on worker nodes
- **Ideally: Containers for isolation and orchestration, but not for distribution**

Working Set

- $\approx$2 % to 10 % of all available files are requested at runtime
- Median of file sizes: $< 4\,\mathrm{kB}$

Flash Crowd Effect

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



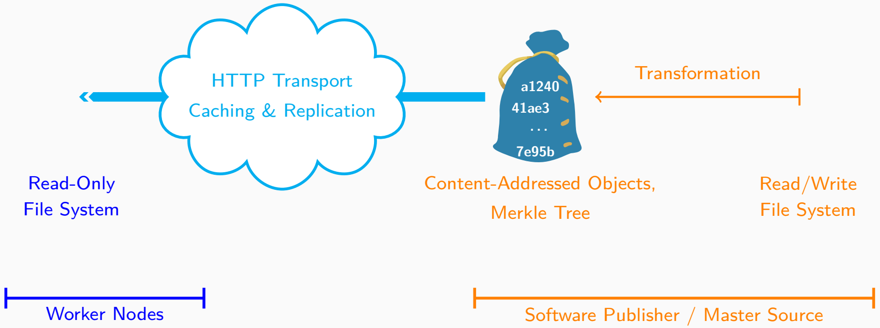
Software vs. Data

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

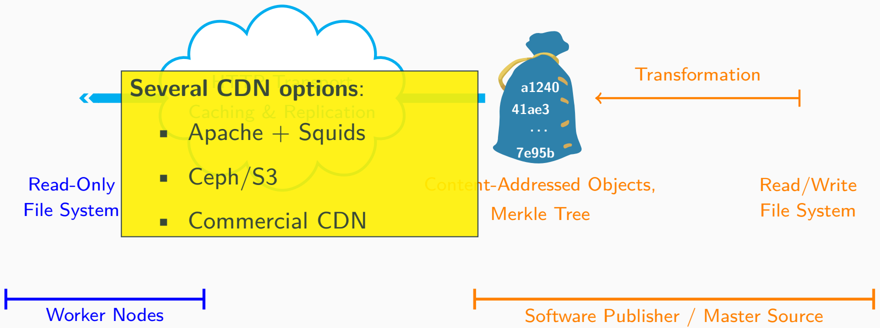
CernVM-FS: A Purpose-Built Software File System

Design Objectives



1. World-wide scalability
2. Infrastructure compatibility
3. Application-level consistency
4. Efficient meta-data access

Design Objectives

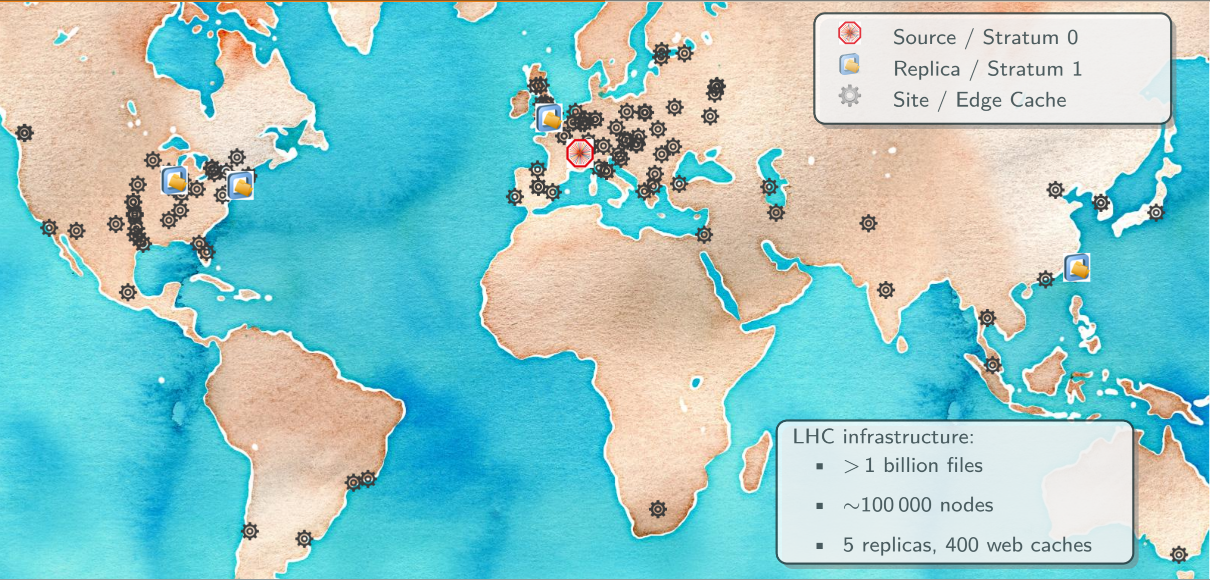


Several CDN options:

- Apache + Squids
- Ceph/S3
- Commercial CDN

1. World-wide scalability
2. Infrastructure compatibility
3. Application-level consistency
4. Efficient meta-data access

Scale of Deployment

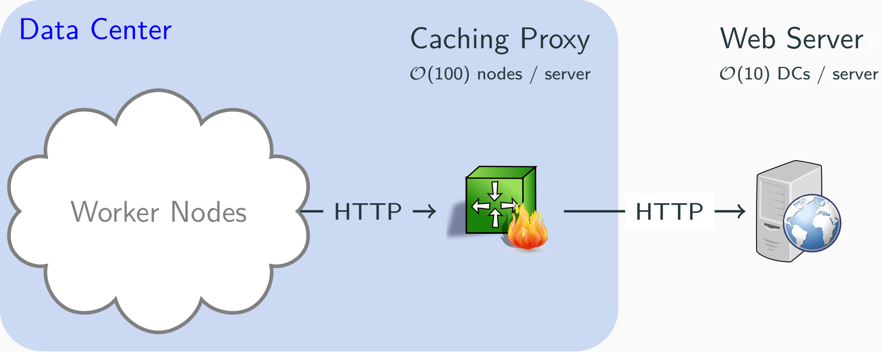


LHC infrastructure:

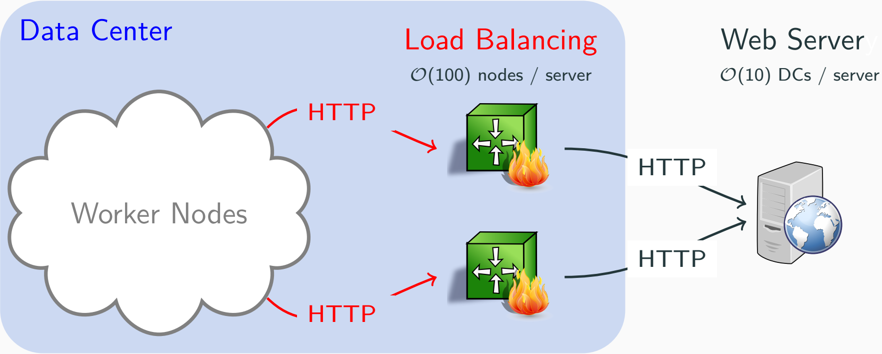
- >1 billion files
- $\sim$100 000 nodes
- 5 replicas, 400 web caches

High-Availability by Horizontal Scaling

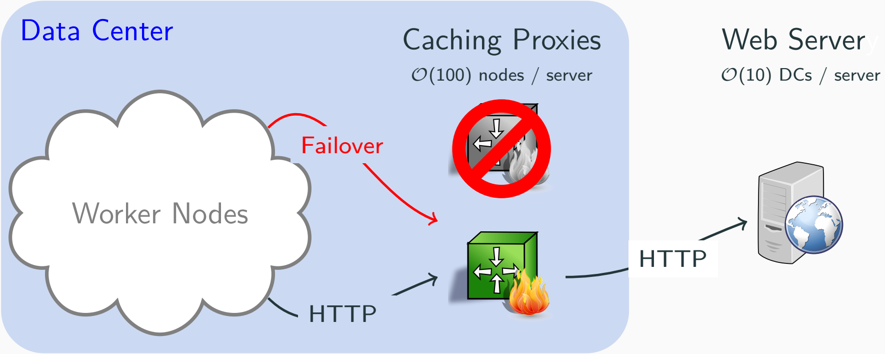
Server side: stateless services



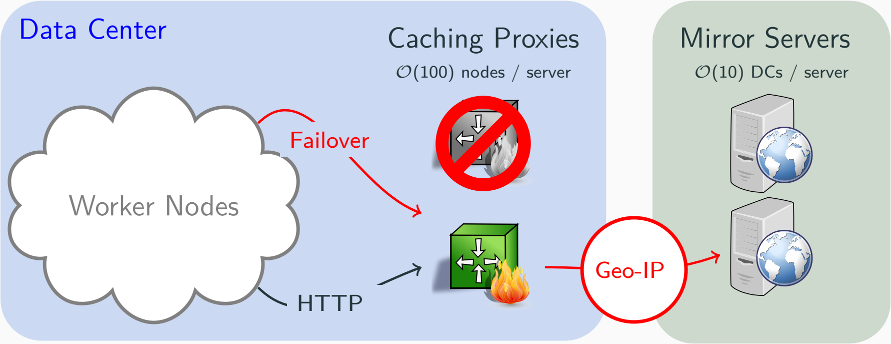
Server side: stateless services



Server side: stateless services

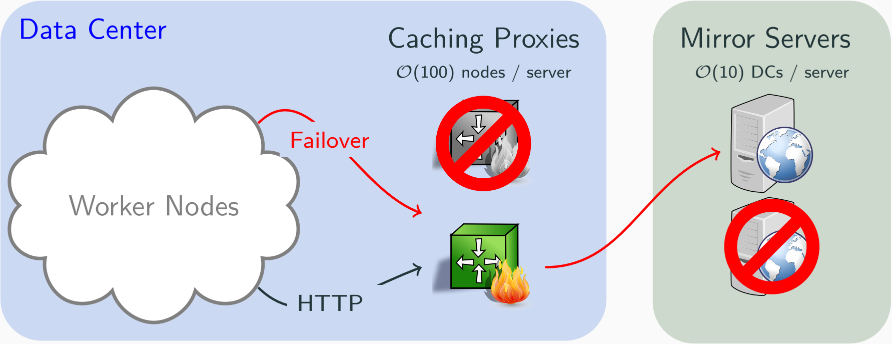


Server side: stateless services

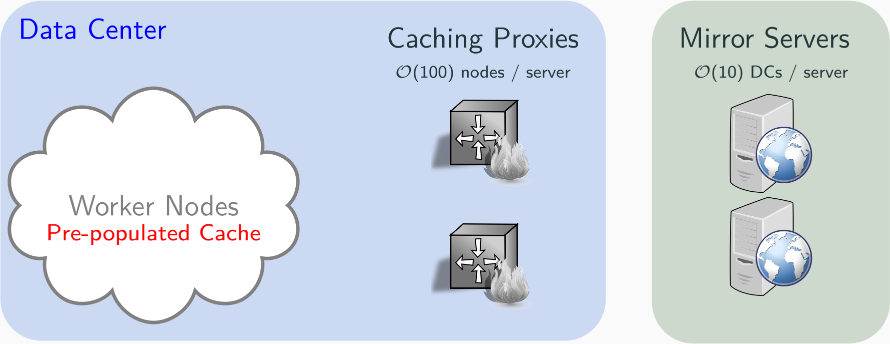


High-Availability by Horizontal Scaling

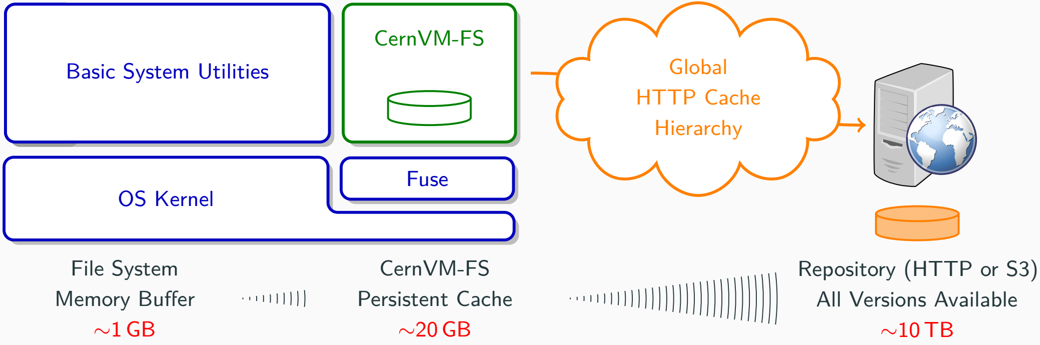
Server side: stateless services



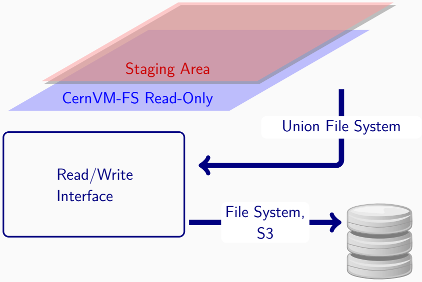
Server side: stateless services



Reading



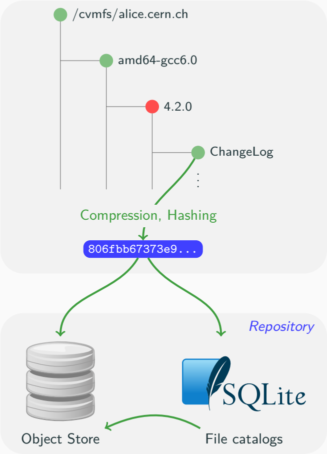
- Fuse based, independent mount points, e. g. /cvmfs/atlas.cern.ch
- High cache effiency because entire cluster likely to use same software



Publishing new content

```
[ ~ ]# cvmfs_server transaction containers.cern.ch
[ ~ ]# cd /cvmfs/containers.cern.ch && 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

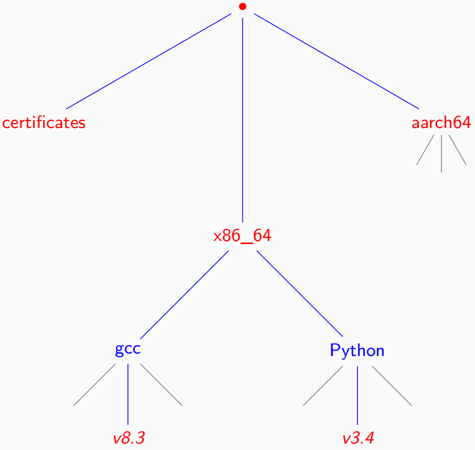
Object Store

- Compressed files and chunks
- De-duplicated

File Catalog

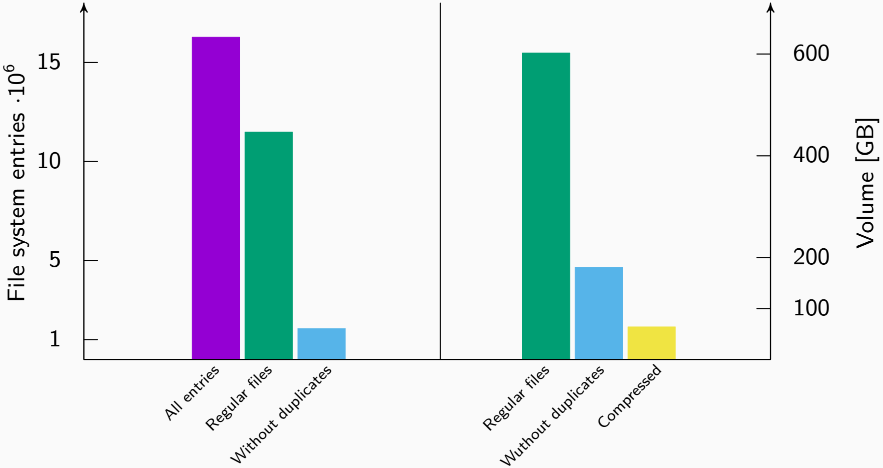
- Directory structure, symlinks
- Content hashes of regular files
- Large files: chunked with rolling checksum
- Digitally signed
- Time to live
- Partitioned / Merkle hashes (possibility of sub catalogs)

Partitioning of Meta-Data



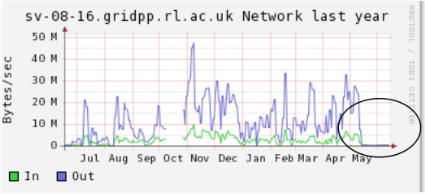
- Locality by software version
- Locality by frequency of changes
- Partitioning up to software librarian, steering through `.cvmfscatalog` magical marker files

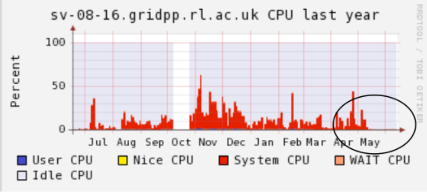
24 months of software releases for a single LHC experiment



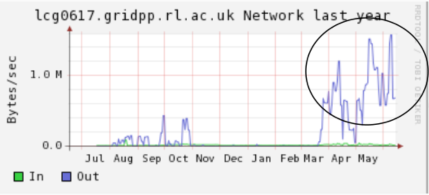
Site-local network traffic: CernVM-FS compared to NFS

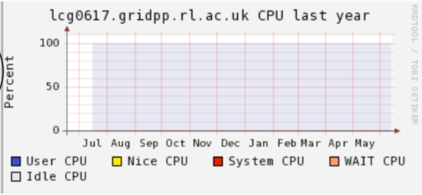
NFS server before and after the switch:





Site squid web cache before and after the switch:

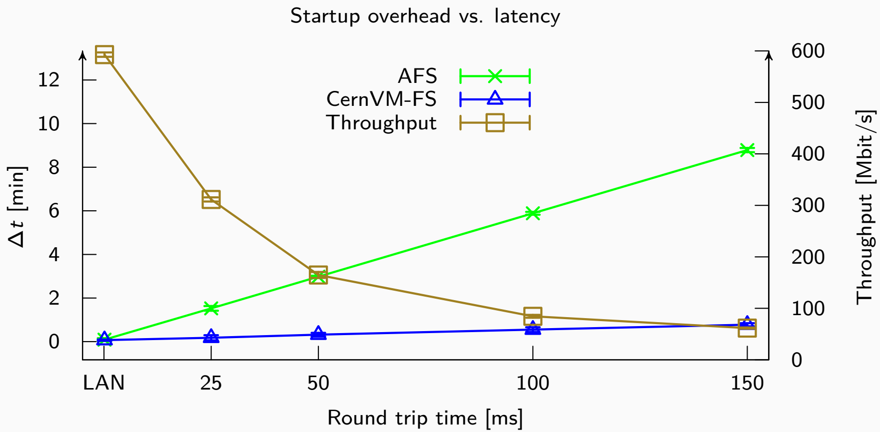




Source: Ian Collier

Latency sensitivity: CernVM-FS compared to AFS

Use case: starting "stressHepix" standard benchmark



Principal Application Areas

❶ Production Software

Example: */cvmfs/ligo.egi.eu*

- ✓ Most mature use case
- Fully unprivileged deployment of fuse module

❷ Integration Builds

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

- ✓ High churn, requires regular garbage collection
- Update propagation from minutes to seconds

❸ Unpacked Container Images

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

- ✓ Works out of the box with Singularity
- ✓ CernVM-FS driver for Docker
- Integration with containerd / kubernetes

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

Summary

- CernVM-FS: special-purpose virtual file system that provides a global shared software area for many scientific collaborations
- Content-addressed storage and asynchronous writing (*publishing*) key to meta-data scalability
- Current areas of development:
  - Fully unprivileged deployment
  - Integration with containerd/kubernetes image management engine

https://github.com/cvmfs/cvmfs

Backup Slides

Links

Source code:	https://github.com/cvmfs/cvmfs
Downloads:	https://cernvm.cern.ch/portal/filesystem/downloads
Documentation:	https://cvmfs.readthedocs.org
Mailing list:	cvmfs-talk@cern.ch
JIRA bug tracker:	https://sft.its.cern.ch/jira/projects/CVM

Supported Platforms

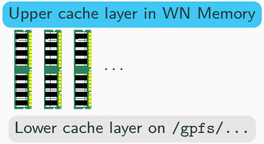
- *A* Platforms:
  - EL 5–7 (soon: 8) AMD64
  - Ubuntu 16.04, 18.04 AMD64
- *B* Platforms
  - macOS, latest two versions
  - SLES 11 – 12
  - Fedora, latest two versions
  - Debian 8–9
  - Ubuntu 12.04 and 14.04
  - EL7 AArch64
  - IA32 architecture
- Experimental: POWER, Raspberry Pi, RISC-V
- Blue sky idea: Windows based on ProjFS

Based on the current needs.
Any platform with Fuse support should be straight-forward to address.

CernVM-FS à la Carte

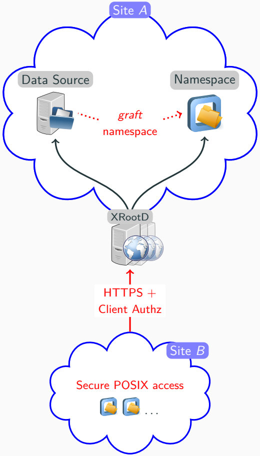
❶ HPC Client Deployment

- Piz Daint: Europe's fastest supercomputer
- Runs LHC experiment jobs from native Fuse client



▸ M Gila (CSCS)

❷ Data Namespace: */cvmfs/*.osgstorage.org*



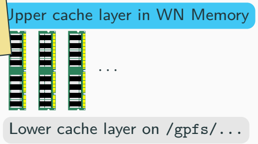
CernVM-FS à la Carte

❶ HPC Client Deployment

Custom client cache:
hierarchical cache
+ RAM disk plugin

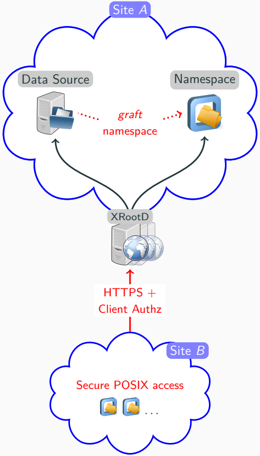
...mputer

- Runs LHC experiment jobs from native Fuse client



▸ M Gila (CSCS)

❷ Data Namespace: */cvmfs/*.osgstorage.org*

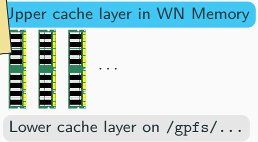


CernVM-FS à la Carte

❶ HPC Client Deployment

Custom client cache:
hierarchical cache
+ RAM disk plugin

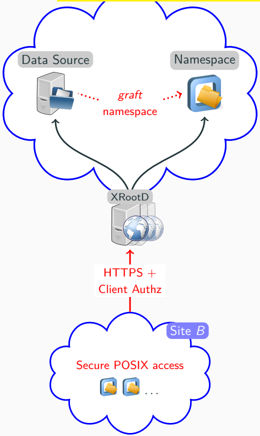
- Runs LHC experiment jobs from native Fuse client



▸ M Gila (CSCS)

❷ Data Namespace: */cvmfs/*.osgstorage.org*

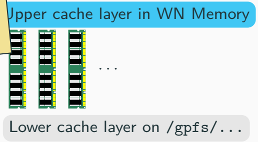
access to multiple PB of data



CernVM-FS à la Carte

❶ HPC Client Deployment

Custom client cache:
hierarchical cache
+ RAM disk plugin

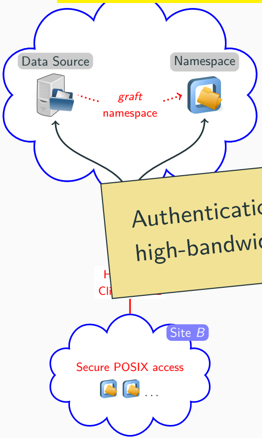


▸ M Gila (CSCS)

- Runs LHC experiment jobs from native Fuse client

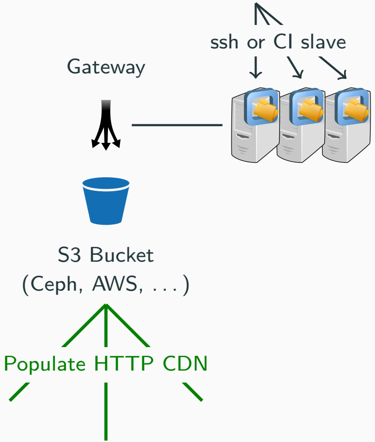
❷ Data Namespace: */cvmfs/*.osgstorage.org*

access to multiple PB of data



Authentication plugin +
high-bandwidth CDN

Distributed Publishing

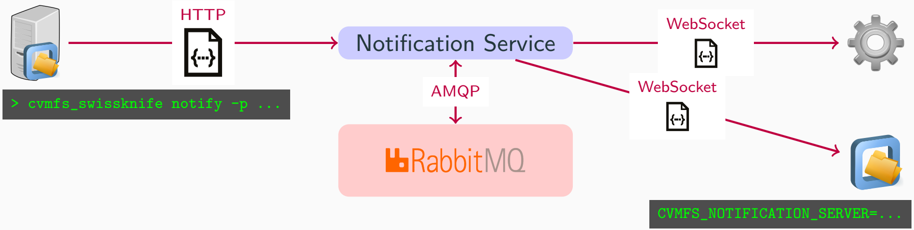


Coordinating Multiple Publisher Nodes

- 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**

Notification Service

Fast distribution channel for repository manifest: useful for CI pipelines, data QA



- Optional service supporting a regular repository
- Subscribe component integrated with the client, automatic reload on changes

→ **CernVM-FS writing remains asynchronous but with fast response time in $\mathcal{O}$(seconds)**

Unpacked Container Images in CernVM-FS

CernVM-FS Container Integration

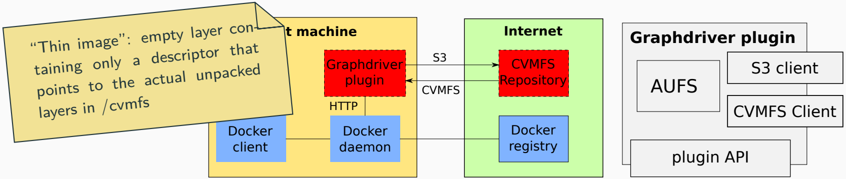
- **Goal**: avoid network congestion by starting unpacked containers from CernVM-FS
- Client / worker node: requires CernVM-FS plug-ins for
  - Docker (available)
  - containerd (in contact with upstream developers)
- CernVM-FS repository: efficient publishing of containers

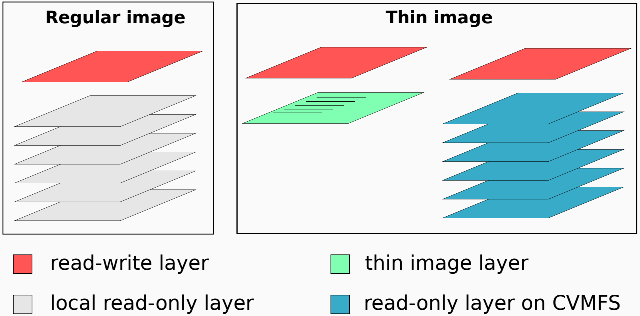
Container Publishing Service

Add-on service on the publisher node to facilitate container conversion from a Docker registry

▸ Presentation

Custom Docker Graph Driver I

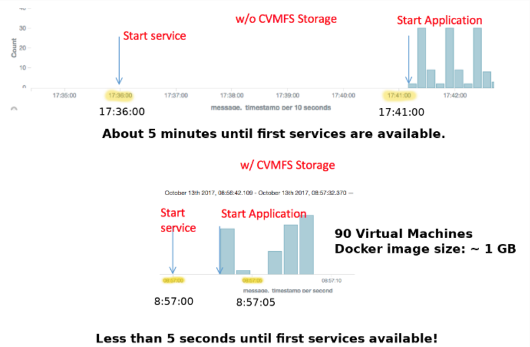




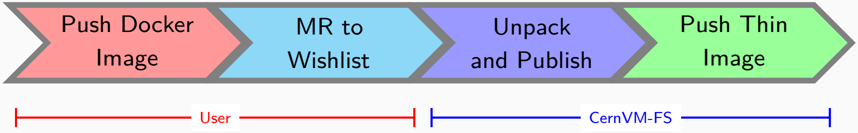
Custom Docker Graph Driver II

```
λ docker plugin install cvmfs/graphdriver
λ docker run cvmfs/thin_cernvm "echo Hello, World!"
```

See https://cvmfs.readthedocs.io/en/latest/cpt-graphdriver.html



Container Publishing Service: Workflow



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:latest'
```

/cvmfs/unpacked.cern.ch

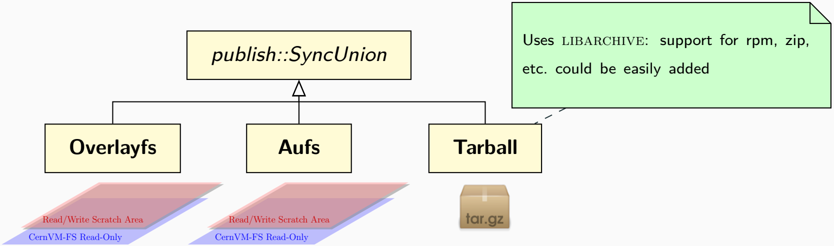
```
# Singularity
/registry.hub.docker.com/fedora:latest -> \
  /cvmfs/unpacked.cern.ch/.flat/d0/d0932...
# Docker with thin image
/.layers/f0/1af7...
```

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
```

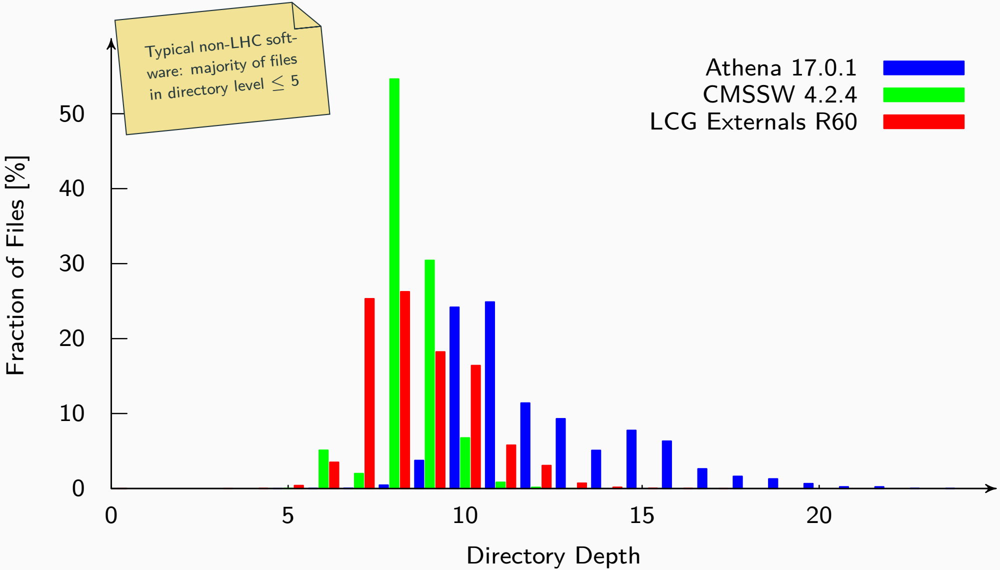
```
$ zcat ubuntu.tar.gz | \
    cvmfs_server ingest -t -
```



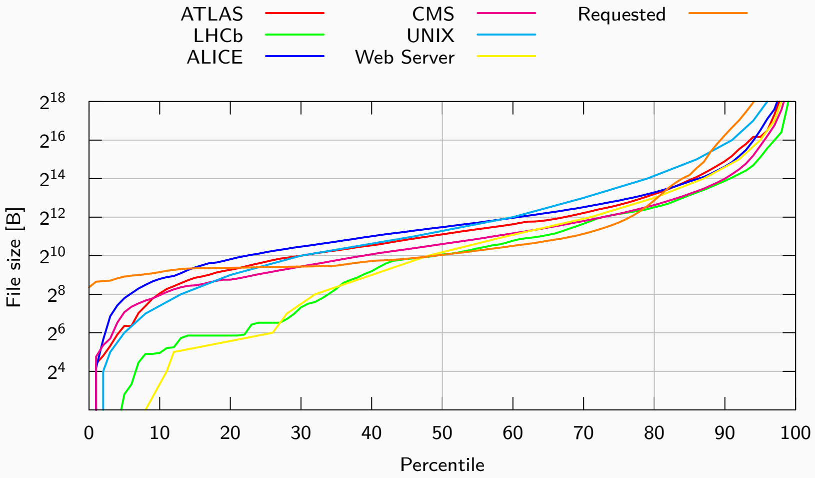
Performance Example

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

Directory Organization

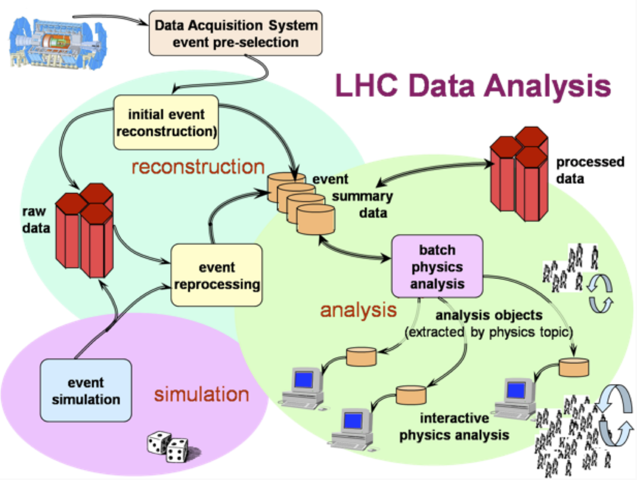


Cumulative File Size Distribution



cf. Tanenbaum et al. 2006 for “Unix” and “Webserver”

LHC Data Flow



Source: Harvey et al.

Selected publications i

- Blomer, J., Aguado-Sanchez, C., Buncic, P., and Harutyunyan, A. (2011).
  Distributing LHC application software and conditions databases using the CernVM file system.
  Journal of Physics: Conference Series, 331(042003).
- Blomer, J., Buncic, P., Meusel, R., Ganis, G., Sfiligoi, I., and Thain, D. (2015).
  The evolution of global scale filesystems for scientific software distribution.
  Computing in Science and Engineering, 17(6):61–71.
- Weitzel, D., Bockelman, B., Dykstra, D., Blomer, J., and Meusel, R. (2017).
  Accessing data federations with cvmfs.
  Journal of Physics: Conference Series, 898(062044).

Selected publications ii

- Blomer, J., Ganis, G., Hardi, N., and Popescu, R. (2017).
  ***Delivering LHC Software to HPC Compute Elements with CernVM-FS*, pages 724–730.**
  Number 10524 in Lecture Notes in Computer Science. Springer.
- Hardi, N., Blomer, J., Ganis, G., and Popescu, R. (2018).
  Making containers lazy with docker and CernVM-FS.
  Journal of Physics: Conference Series, 1085.