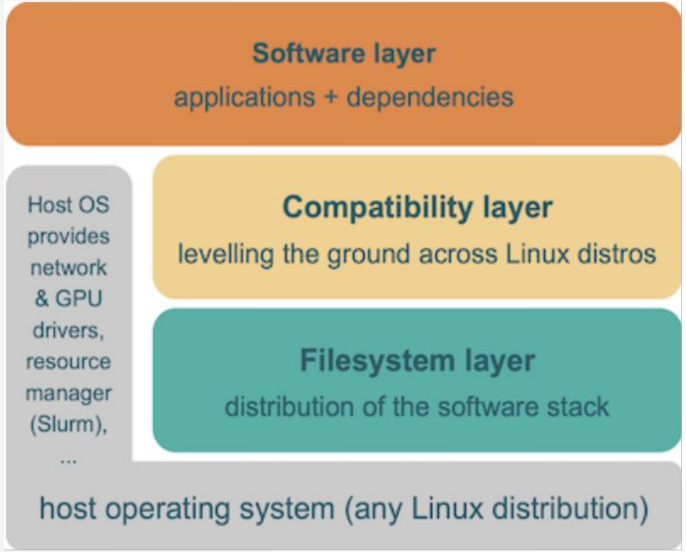
The CernVM FileSystem (CVMFS)

Valentin Völkl
CERN/EP-SFT
CVMFS Developers Team

March 27nd, 2025, Easybuild users workshop



Software layer

applications + dependencies

Compatibility layer

levelling the ground across Linux distros

Filesystem layer

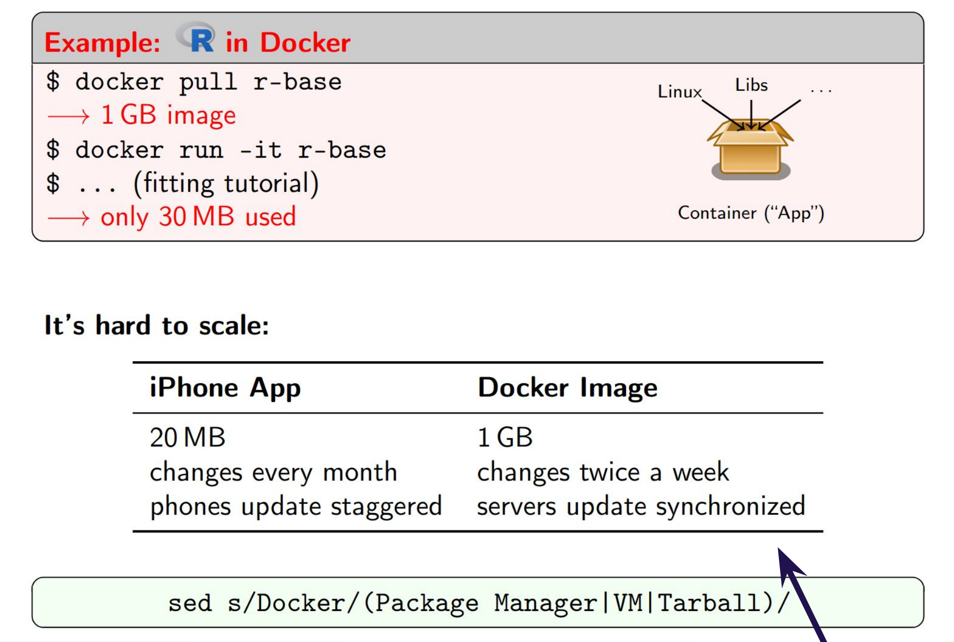
distribution of the software stack

Host OS provides network & GPU drivers, resource manager (Slurm), ...

host operating system (any Linux distribution)

YOU ARE HERE

The problem with software distribution



Example: R in Docker

`$ docker pull r-base`

⟶ 1 GB image

`$ docker run -it r-base`

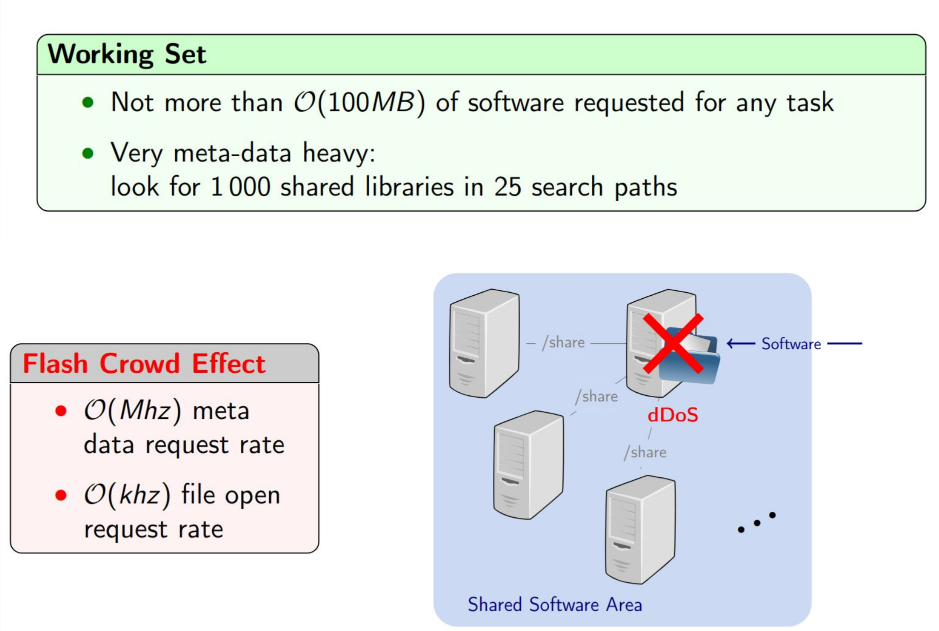
`$ ...` (fitting tutorial)

⟶ only 30 MB used

It’s hard to scale:

iPhone App	Docker Image
20 MB	1 GB
changes every month	changes twice a week
phones update staggered	servers update synchronized

`sed s/Docker/(Package Manager|VM|Tarball)/`



Working Set

- Not more than $\mathcal{O}(100MB)$ of software requested for any task
- Very meta-data heavy:
  look for 1 000 shared libraries in 25 search paths

Flash Crowd Effect

- $\mathcal{O}(Mhz)$ meta data request rate
- $\mathcal{O}(khz)$ file open request rate

HEP software stacks are even bigger,
and hard to package efficiently!

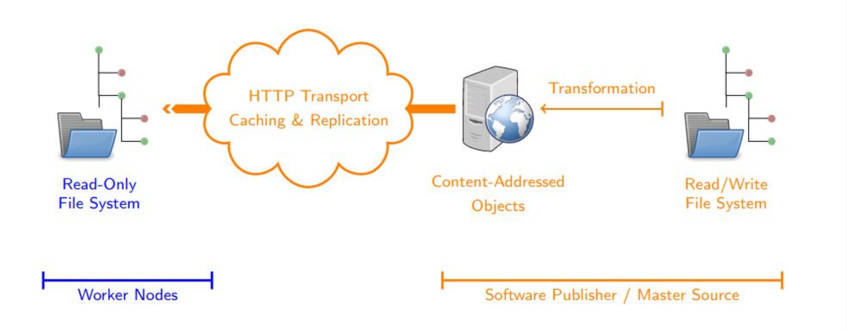
CernVM FileSystem (CVMFS)

- Global, **read-only filesystem** for **software distribution**
  - with a user experience similar to an on-demand streaming service (... but for scientific software)

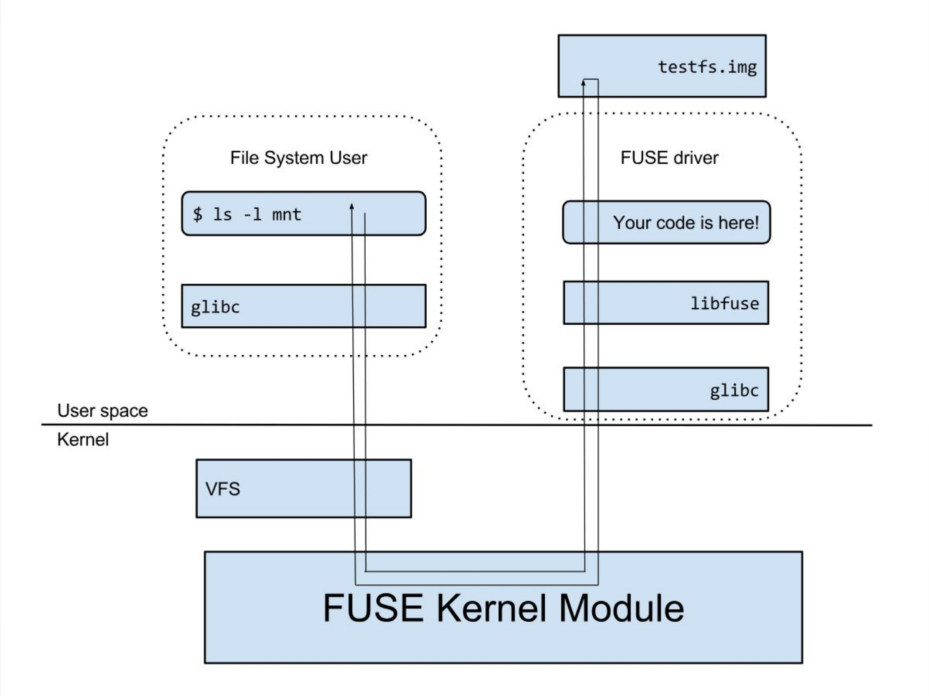
```
~$ ls /cvmfs
~$ ls /cvmfs/software.eessi.io # mounted automatically by autofs
repo
~$ ls /cvmfs/software.eessi.io
host_injections  init  README.eessi  versions
~$ cat /cvmfs/software.eessi.io/README.eessi # just-in-time download
EESSI - the European Environment for Scientific Software Installations

Getting started
---------------
```

CernVM FileSystem (CVMFS)

- Global, **read-only filesystem** for **software distribution**
  - with a user experience similar to an on-demand streaming service (… but for scientific software)
- implemented as a filesystem in userspace, via *libfuse*
  - allows client to be installed flexibly on all workernodes
- Optimized for storing and distributing software
  - Content-adressable storage allows **De-duplication**
  - Multi-level **caching**, use of HTTP transport
  - **Compression** of data
  - Verification of data **integrity**
  - …



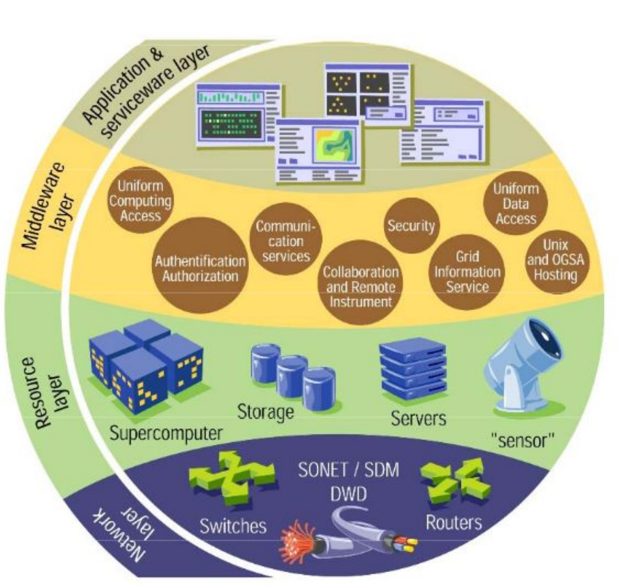
CVMFS is a Filesystem in Userspace

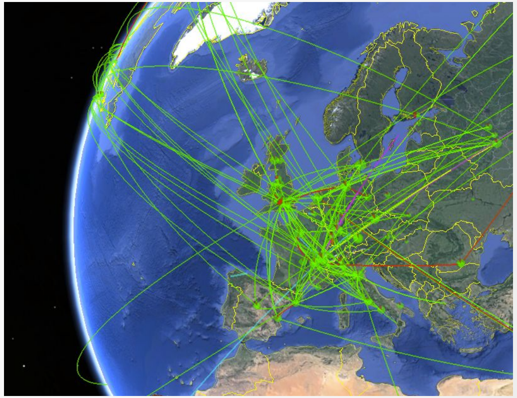


- Implements all necessary (ro) syscalls
- If file is in local cache: use that
- If file is not in local cache: download from object store and place it in local cache, use that

Originally developed for LHC “Computing Grid”

- Intended to provide uniform computing access to all resources pledged for the LHC



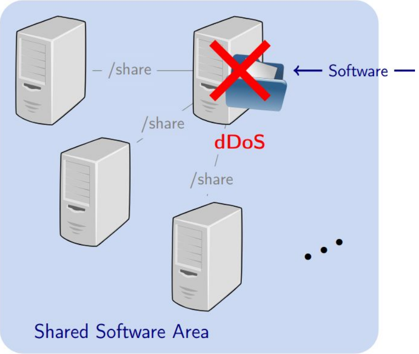


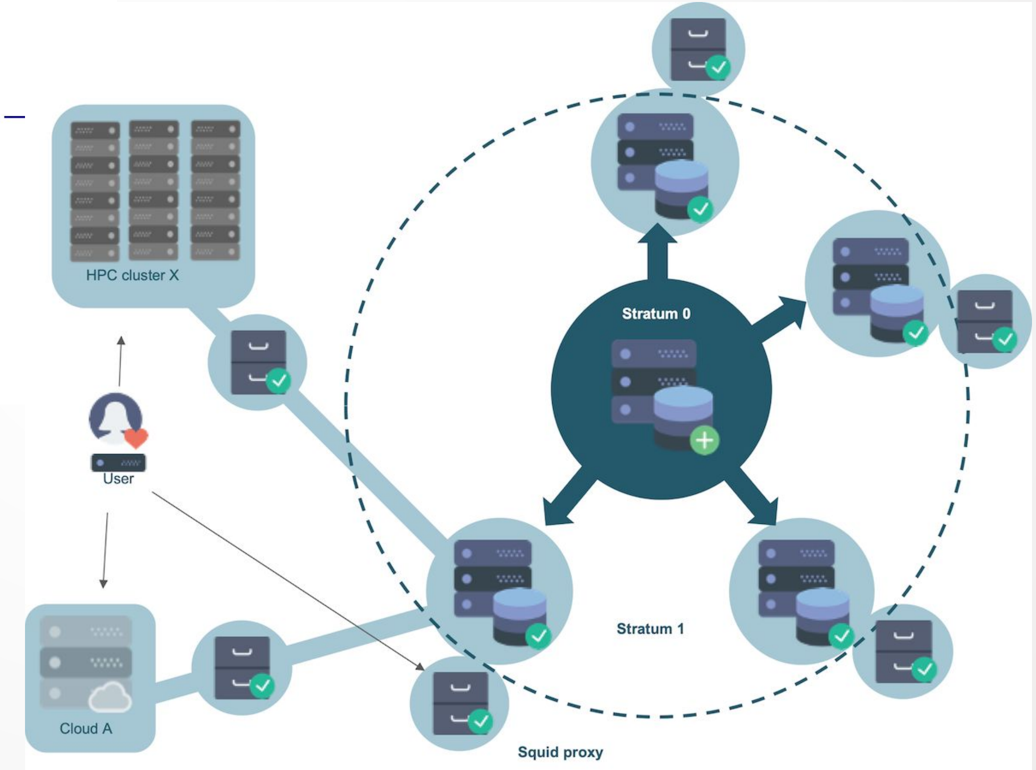
Working Set

- Not more than $\mathcal{O}(100MB)$ of software requested for any task
- Very meta-data heavy:
  look for 1 000 shared libraries in 25 search paths

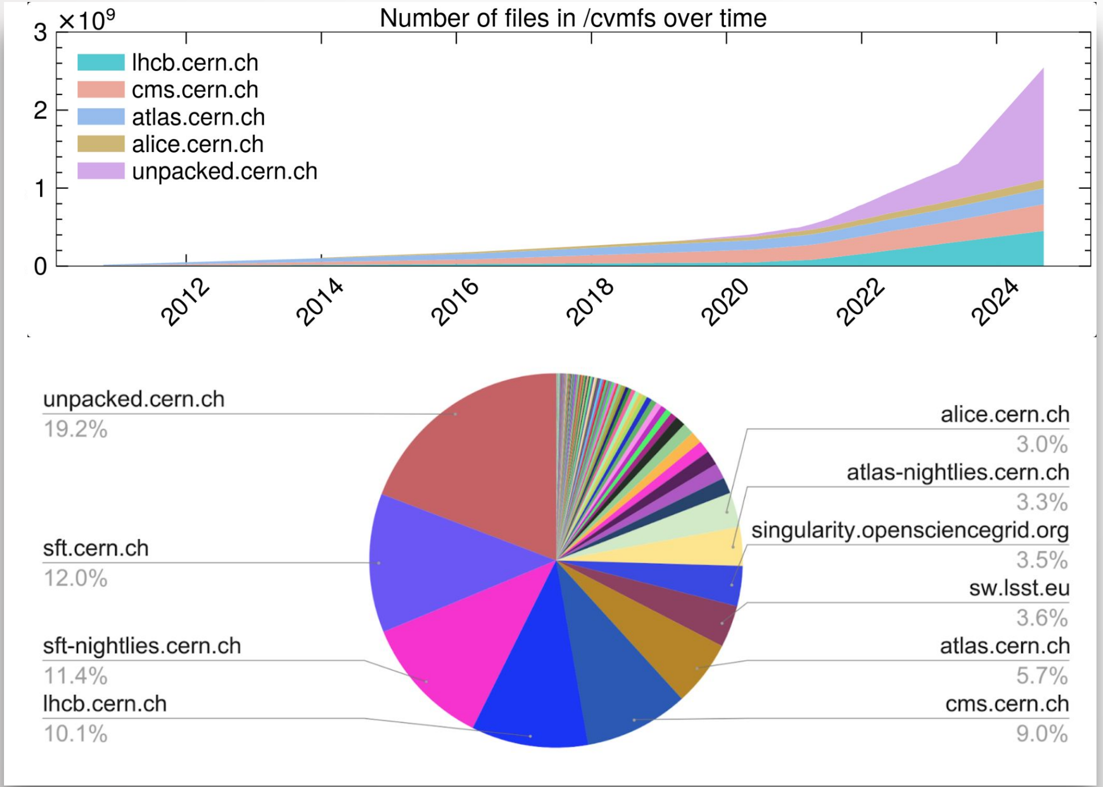
Flash Crowd Effect

- $\mathcal{O}(Mhz)$ meta data request rate
- $\mathcal{O}(khz)$ file open request rate





CVMFS (at CERN) in numbers



~ **15 Stratum 1s**

~ **> 4 B files** in the /cvmfs tree

~ **2 PB of data** accessible through /cvmfs

out of which ~1.5 PB in *external* files

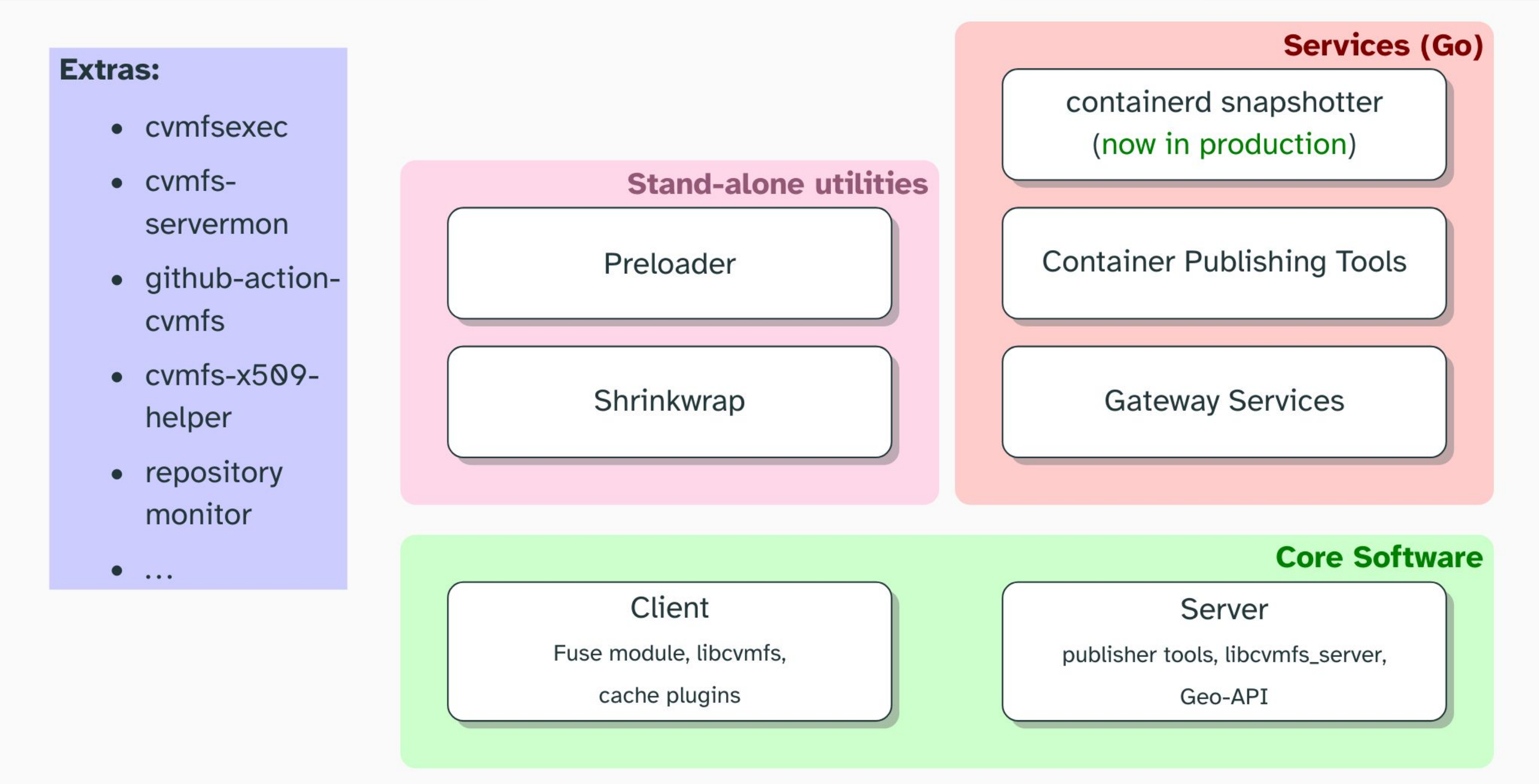
proven to scale up to 100 PB

~ **> 4k container images**

~ **260 repositories**

- Backed by S3(+CEPH) or local storage

CernVM-FS Code and components



Extras:

- cvmfsexec
- cvmfs-servermon
- github-action-cvmfs
- cvmfs-x509-helper
- repository monitor
- …

Stand-alone utilities

Preloader

Shrinkwrap

Services (Go)

containerd snapshotter (now in production)

Container Publishing Tools

Gateway Services

Core Software

Client

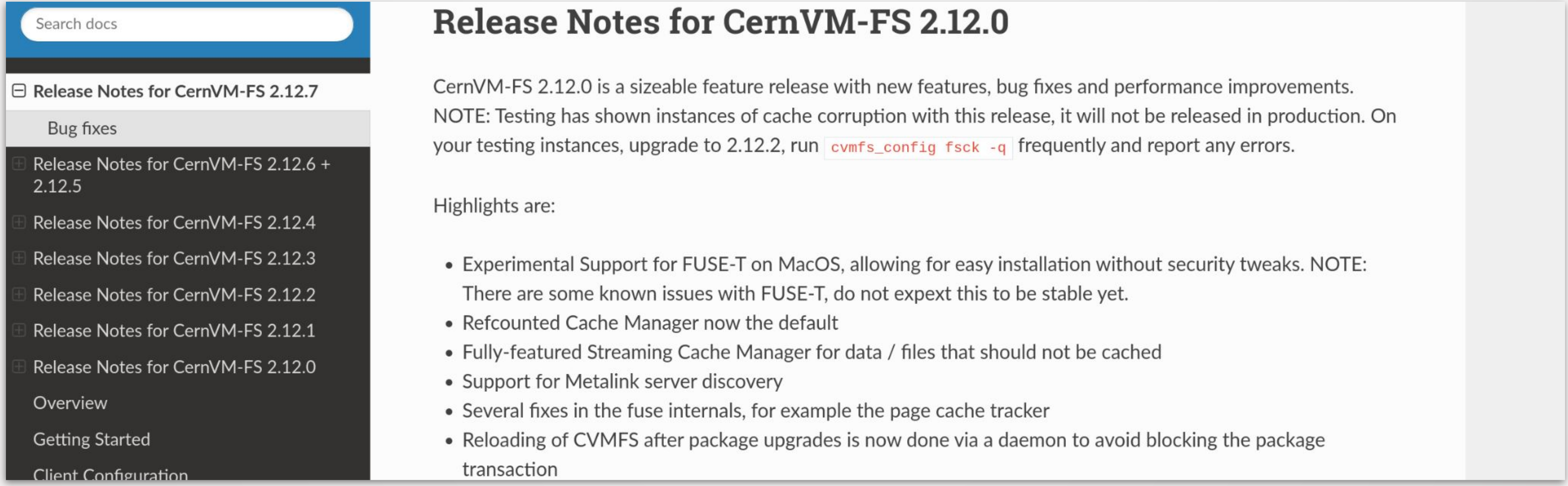
Fuse module, libcvmfs, cache plugins

Server

publisher tools, libcvmfs_server, Geo-API

Current status: CVMFS 2.12.7 (Released 2025)

See full changelog for more details:


Search docs
Release Notes for CernVM-FS 2.12.7
Bug fixes
Release Notes for CernVM-FS 2.12.6 + 2.12.5
Release Notes for CernVM-FS 2.12.4
Release Notes for CernVM-FS 2.12.3
Release Notes for CernVM-FS 2.12.2
Release Notes for CernVM-FS 2.12.1
Release Notes for CernVM-FS 2.12.0
Overview
Getting Started
Client Configuration

Release Notes for CernVM-FS 2.12.0

CernVM-FS 2.12.0 is a sizeable feature release with new features, bug fixes and performance improvements. NOTE: Testing has shown instances of cache corruption with this release, it will not be released in production. On your testing instances, upgrade to 2.12.2, run `cvmfs_config fsck -q` frequently and report any errors.

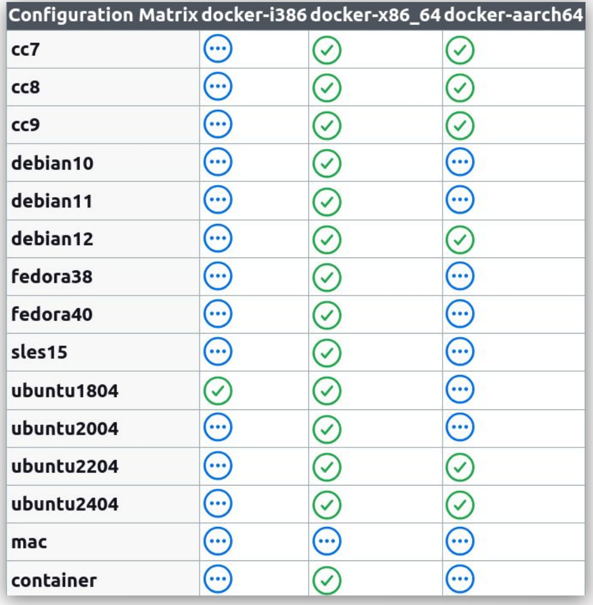
Highlights are:

- Experimental Support for FUSE-T on MacOS, allowing for easy installation without security tweaks. NOTE: There are some known issues with FUSE-T, do not expext this to be stable yet.
- Refcounted Cache Manager now the default
- Fully-featured Streaming Cache Manager for data / files that should not be cached
- Support for Metalink server discovery
- Several fixes in the fuse internals, for example the page cache tracker
- Reloading of CVMFS after package upgrades is now done via a daemon to avoid blocking the package transaction

Packaging

```
yum install cvmfs
apt install cvmfs
```

- Providing pre-built packages and yum/apt repositories seems to be appreciated
- cvmfs-prod and cvmfs-testing
  - Plan to add cvmfs-devel
- Target firstly:
  - RHEL(-clones) and Debian
  - MacOS for the laptop usecase (no server tools).
  - Open to adding new ones!
- Goal: get packages into upstream repositories for Debian and Fedora



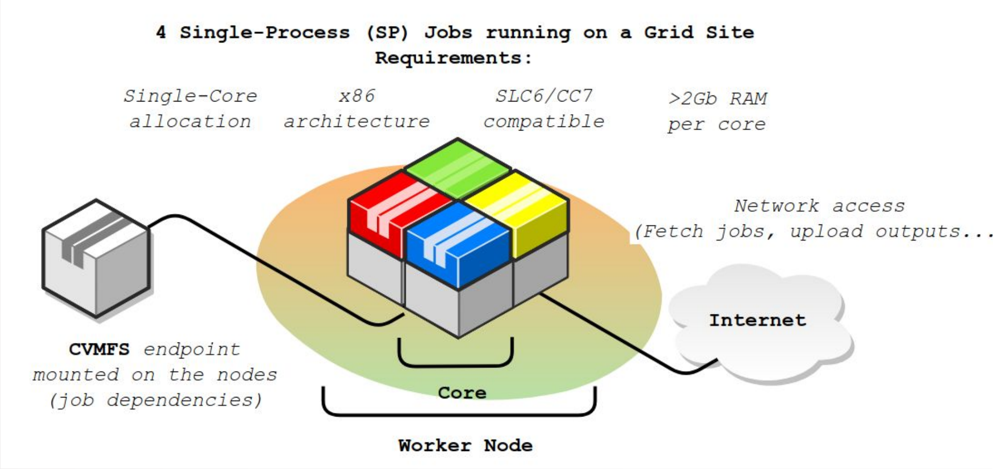
Configuration Matrix	docker-i386	docker-x86_64	docker-aarch64
cc7	⋯	✓	✓
cc8	⋯	✓	✓
cc9	⋯	✓	✓
debian10	⋯	✓	⋯
debian11	⋯	✓	⋯
debian12	⋯	✓	✓
fedora38	⋯	✓	⋯
fedora40	⋯	✓	⋯
sles15	⋯	✓	⋯
ubuntu1804	✓	✓	⋯
ubuntu2004	⋯	✓	⋯
ubuntu2204	⋯	✓	✓
ubuntu2404	⋯	✓	✓
mac	⋯	⋯	⋯
container	⋯	✓	⋯

Please do consider using also the `cvmfs-testing` repositories!

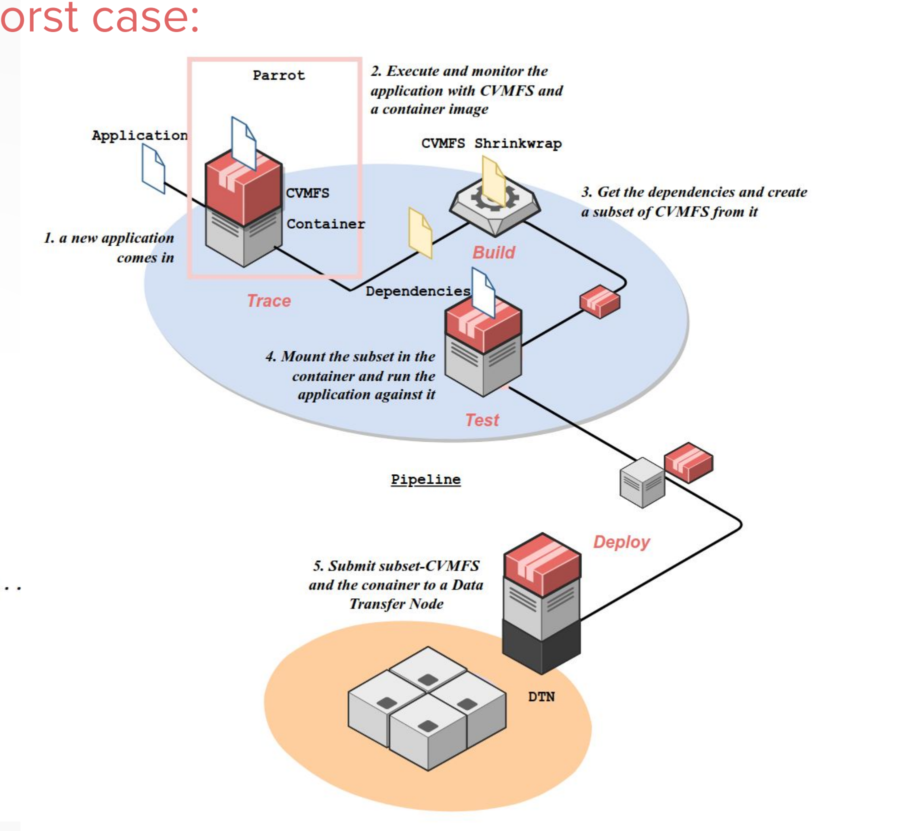
CVMFS on HPC

HPC sites may impose many restrictions. Workarounds for many configurations exist, but come at different levels of cost

Best case:


4 Single-Process (SP) Jobs running on a Grid Site
Requirements:
Single-Core allocation
x86 architecture
SLC6/CC7 compatible
>2Gb RAM per core
Network access (Fetch jobs, upload outputs...
Internet
CVMFS endpoint mounted on the nodes (job dependencies)
Core
Worker Node

Worst case:


Parrot
2. Execute and monitor the application with CVMFS and a container image
Application
CVMFS Shrinkwrap
CVMFS
Container
3. Get the dependencies and create a subset of CVMFS from it
1. a new application comes in
Build
Trace
Dependencies
4. Mount the subset in the container and run the application against it
Test
Pipeline
Deploy
5. Submit subset-CVMFS and the conainer to a Data Transfer Node
DTN

Alexandre F Boyer [CHEP 2023]

cvmfsexec [Dave Dykstra]

For when you have no admin privileges!

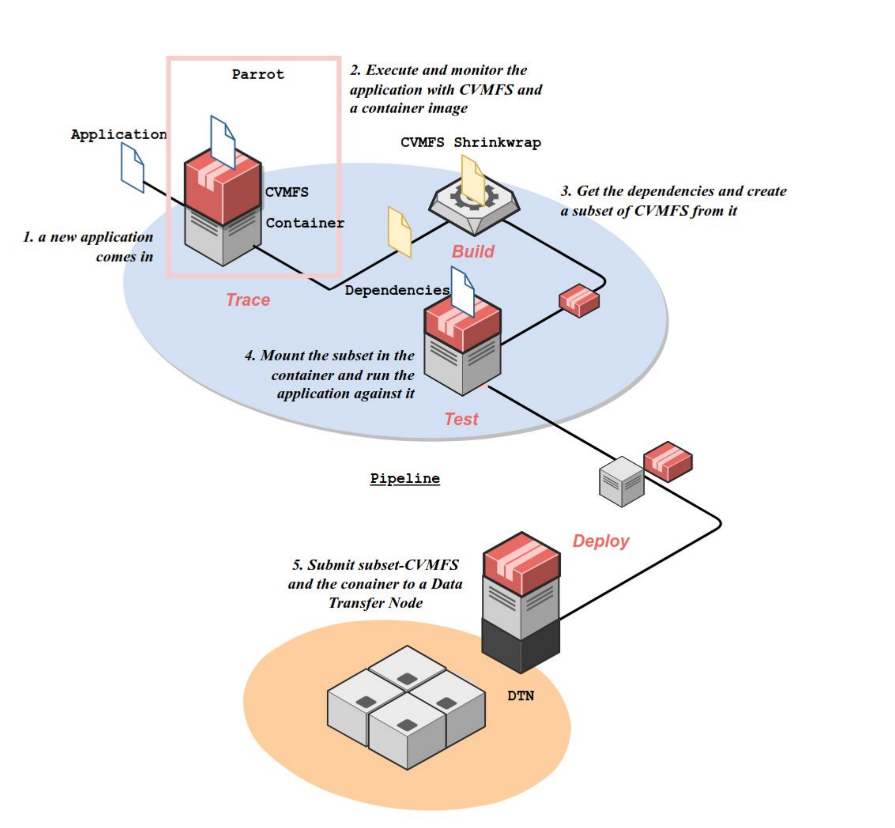
For mounting cvmfs as an unprivileged user, without the cvmfs package being installed by a system administrator.

- 4 modes, depending on availability of certain features on the host
  - Fusermount
  - unprivileged namespace fuse mounts
  - setuid installation of singularity >= 3.4
- - No shared cache on system

CVMFS Shrinkwrap

For when you have no network connection!

- Copies contents of CVMFS to a local file system
  - rsync, but more efficient (keeps deduplication)
- Some experiments have built a whole framework to trace the file accesses of specific jobs and shrinkwrap only that
- Similar cvmfs_preload
- - Labor intensive


Parrot
2. Execute and monitor the application with CVMFS and a container image
Application
CVMFS Shrinkwrap
CVMFS
Container
3. Get the dependencies and create a subset of CVMFS from it
1. a new application comes in
Build
Trace
Dependencies
4. Mount the subset in the container and run the application against it
Test
Pipeline
Deploy
5. Submit subset-CVMFS and the conainer to a Data Transfer Node
DTN

Advanced cache configurations

For when you have no local discs!

- Use Loopback file system on cache
  - One file per repository
  - Easier on the metadata servers of cluster file system
- Use RAM cache or Tiered Cache
  - Example: https://cvmfs.readthedocs.io/en/stable/cpt-configure.html#example

- Workarounds no longer recommended:
  - NFS exports
    - Some HPC sites have tried running the cvmfs client on just one server and exporting to worker nodes over NFS. These installations can be made to work, but they are very inefficient, and often run into operational problems.
  - Parrot Connector

Proxy / Stratum - 1 infrastructure

- Usual Site Recomendation: Stratum-1 (if possible) + Proxies as needed
- SQUID is default recommendation as it is the production workhorse in HEP.
  - For historical reasons, these proxies also need to work for the “FRONTIER” databases
  - Is a forward proxy by design
- VARNISH Cache under investigation (link to working vcl)
  - Nginx has been used as well

- New usecase with EESSI: partial stratum-1s
  - Replication with respect to architecture
- Will need some careful design
  - Usually stratum 1s can be converted to stratum0s
- But planned to be implemented this year

Licenced Software / Private Repository

- Legal situation around distributing binaries of licenced software on CVMFS not clear

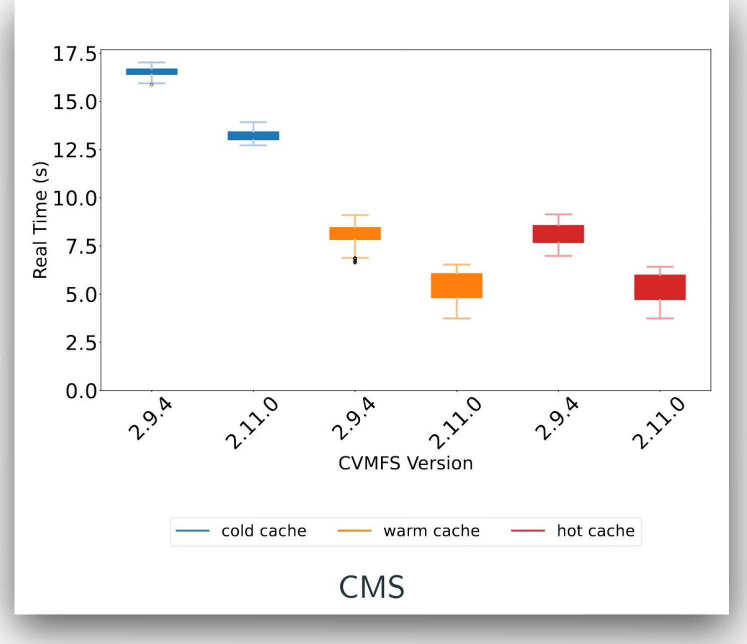
- x509 authenticated repositories are not recommended and will be deprecated

- Approach take at CERN: dedicated repository (projects.cern.ch) that is only available on internal network

Further topics

Performance improvements

- Page Cache Tracker: Much better use of **kernel page cache** (already in 2.10)
- CVMFS_SYMLINK_CACHE possible on new enough FUSE/Kernel versions
  - Requires libfuse 3.10+
  - And kernel in rhel8+
- **Statfs** caching
- CVMFS_USE_REFCOUNT
  - for many-core nodes
- Coming up: More memory stability with improved inode invalidation in the kernel
- Bundled file download for snappier interactive usage

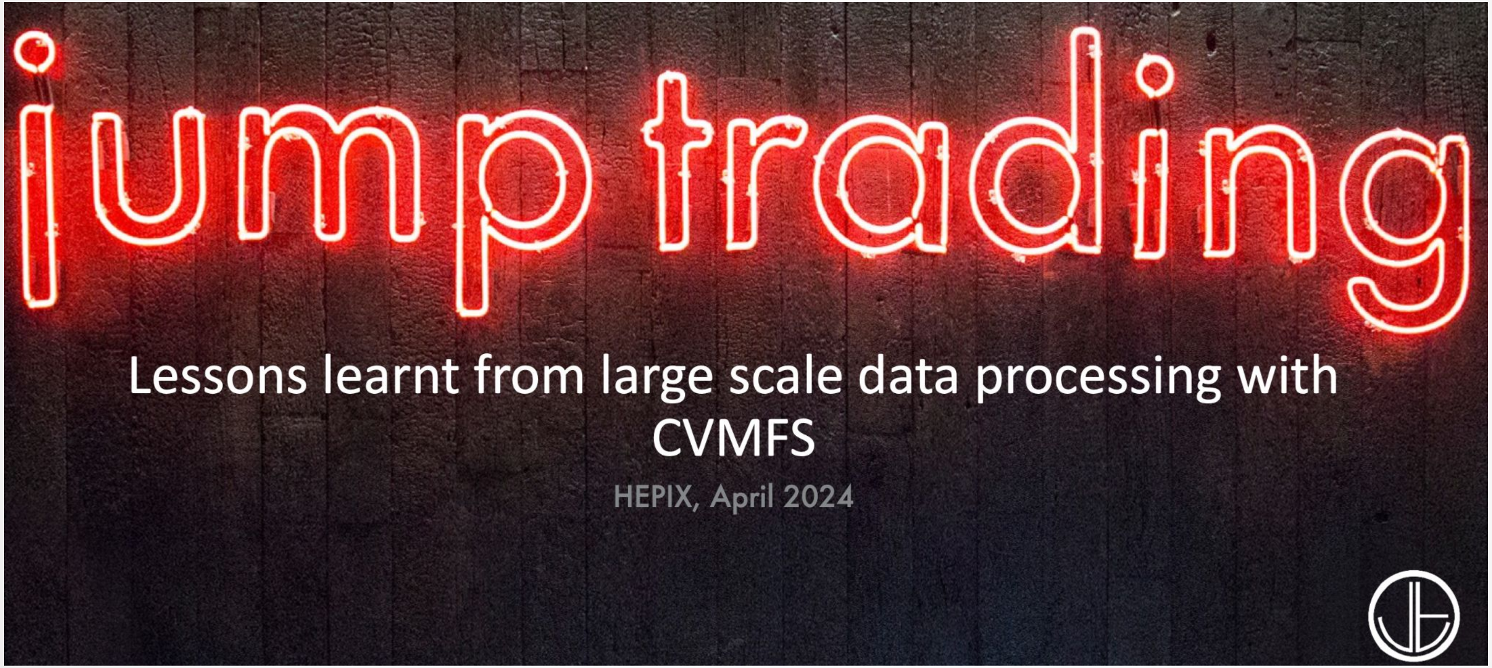


See CHEP 2023 for more details

Jump Trading / Data on CVMFS

CVMFS used as a POSIX Filesystem view on Petabytes of Data.

See Matt Harveys presentation for more details



Lessons learnt from large scale data processing with CVMFS

HEPIX, April 2024

MacOS

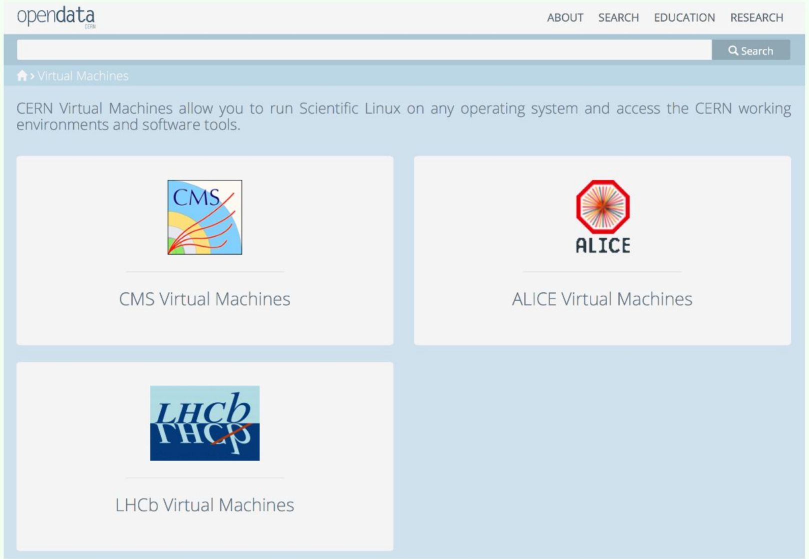
- MacFUSE library stable, but requires kernel extensions enabled
  - More difficult to do with each update, currently 3 reboots
- FUSE-T is an alternative implementation using an NFS-server
  - Less mature, and less performant, but without installation procedure
  - can be used on github actions shared runners
- GSoC project in 2024 evaluated FUSE-T
  - finding few non-critical peculiarities, fixing one critical bug (in FUSE-T)
  - a major blocker (wrong directory listings) remains
  - but homebrew infrastructure, m1 builds in place

- FSKit will likely shake up filesystem APIs again, requiring further attention

CernVM

Currently used mostly for outreach, education and **data/sw preservation**

CERN OpenData Portal, CERN@School



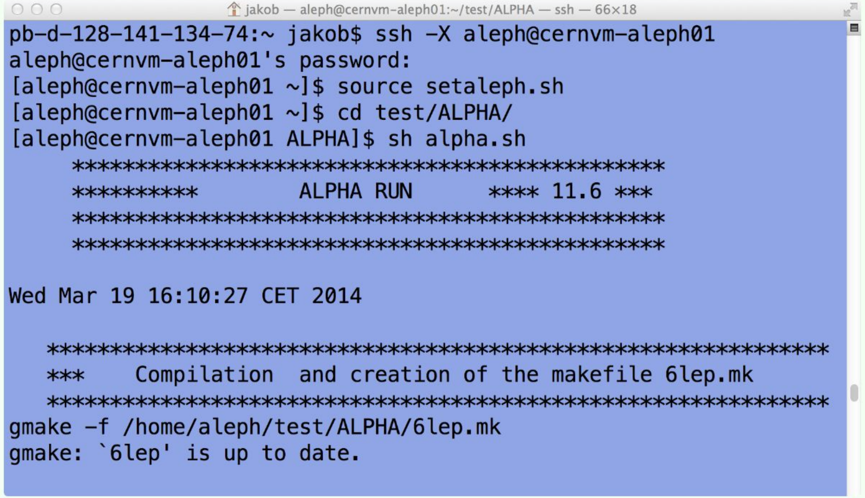
CERN Virtual Machines allow you to run Scientific Linux on any operating system and access the CERN working environments and software tools.

CMS Virtual Machines

ALICE Virtual Machines

LHCb Virtual Machines

ALEPH software in CernVM



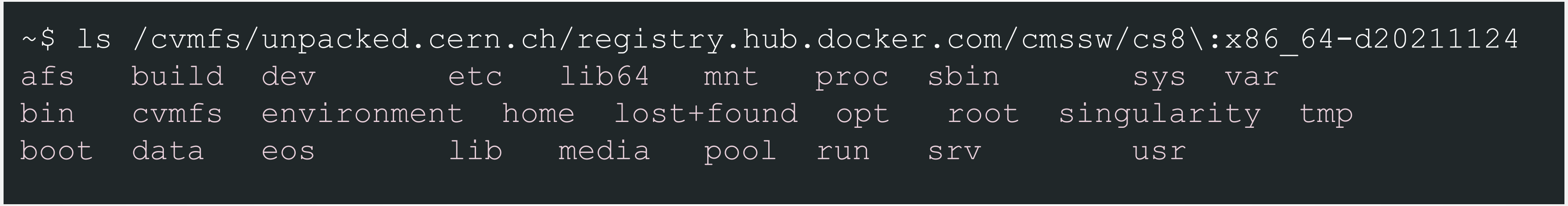
```
pb-d-128-141-134-74:~ jakob$ ssh -X aleph@cernvm-aleph01
aleph@cernvm-aleph01's password:
[aleph@cernvm-aleph01 ~]$ source setaleph.sh
[aleph@cernvm-aleph01 ~]$ cd test/ALPHA/
[aleph@cernvm-aleph01 ALPHA]$ sh alpha.sh
     ***********************************************
     **********         ALPHA RUN      **** 11.6 ***
     ***********************************************
     ***********************************************

Wed Mar 19 16:10:27 CET 2014

   **************************************************************
   ***    Compilation  and creation of the makefile 6lep.mk
   **************************************************************
gmake -f /home/aleph/test/ALPHA/6lep.mk
gmake: `6lep' is up to date.
```

Demonstrates that VMs can bridge 15+ years

- **CernVM-Five**: Container-first implementation, can be useful as base-image+CVMFS!

Containers

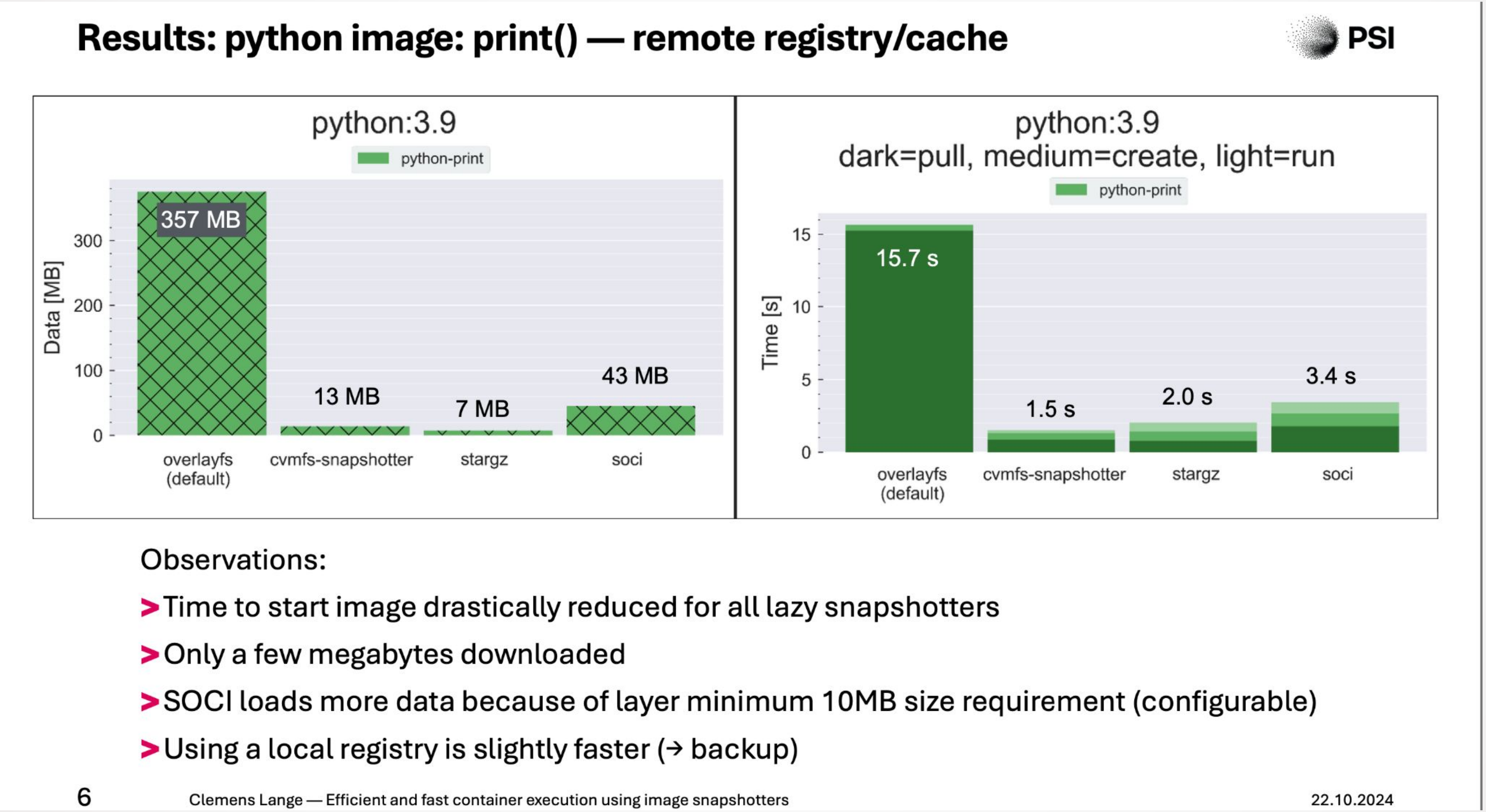
- CVMFS provides tooling to unpack, store and distribute containers, with *unpacked.cern.ch* being the biggest repository:



```
~$ ls /cvmfs/unpacked.cern.ch/registry.hub.docker.com/cmssw/cs8\:x86_64-d20211124
afs   build  dev          etc   lib64   mnt    proc  sbin         sys  var
bin   cvmfs  environment  home  lost+found  opt   root  singularity  tmp
boot  data   eos          lib   media   pool   run   srv          usr
```

- *Apptainer* can directly launch the container from this root file system.
- The same benefits from using CVMFS apply! Leading to:
  - Drastically faster container **startup** times
  - Automatic **cache management** of container images on the worker nodes

Containerd snapshotter



Results: python image: print() — remote registry/cache

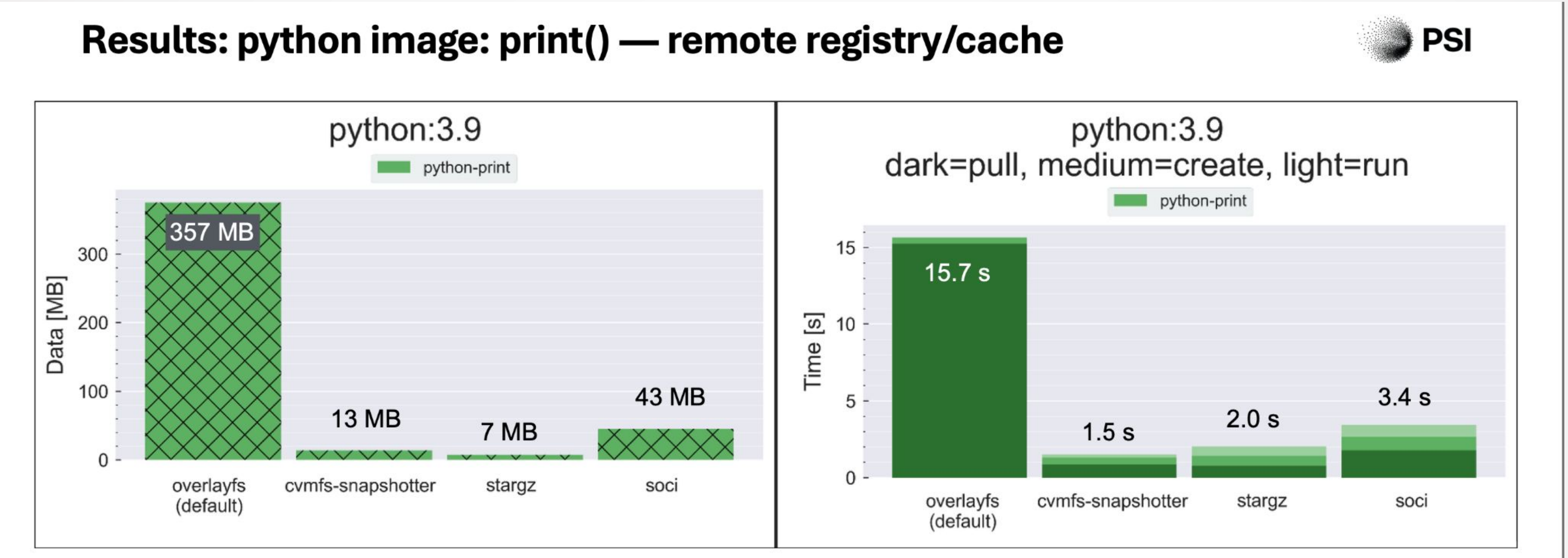
PSI

Observations:

- Time to start image drastically reduced for all lazy snapshotters
- Only a few megabytes downloaded
- SOCI loads more data because of layer minimum 10MB size requirement (configurable)
- Using a local registry is slightly faster (→ backup)

6 Clemens Lange — Efficient and fast container execution using image snapshotters 22.10.2024

Containerd snapshotter



Results: python image: print() — remote registry/cache

Can be used with **ctr, nerdctl, k8s**
and now (since 24.0) **docker** - completely transparently

(layers available on CVMFS are lazy loaded, rest fetched from registry)

Conclusion

- CVMFS lets you stream software on-demand, eessi-ly and efficiently
  - Mature project with long-term support

- Used by High Energy and Nuclear Physics, EESSI,, EUCLID, LIGO, LSST, SKA...
- **Software preservation** built in
- To get started with CVMFS https://multixscale.github.io/cvmfs-tutorial-hpc-best-practices/
  - Website: cernvm.cern.ch
  - Docs: cvmfs.readthedocs.io
- Get in touch (vavolkl@cern.ch, github.com/cvmfs/cvmfs, cernvm-forum.cern.ch)! we are happy to work with you for improvements for the particular situation of your site