# Analysis and Optimization of CVMFS caches on LxPlus

Jakub Ogrodowczyk

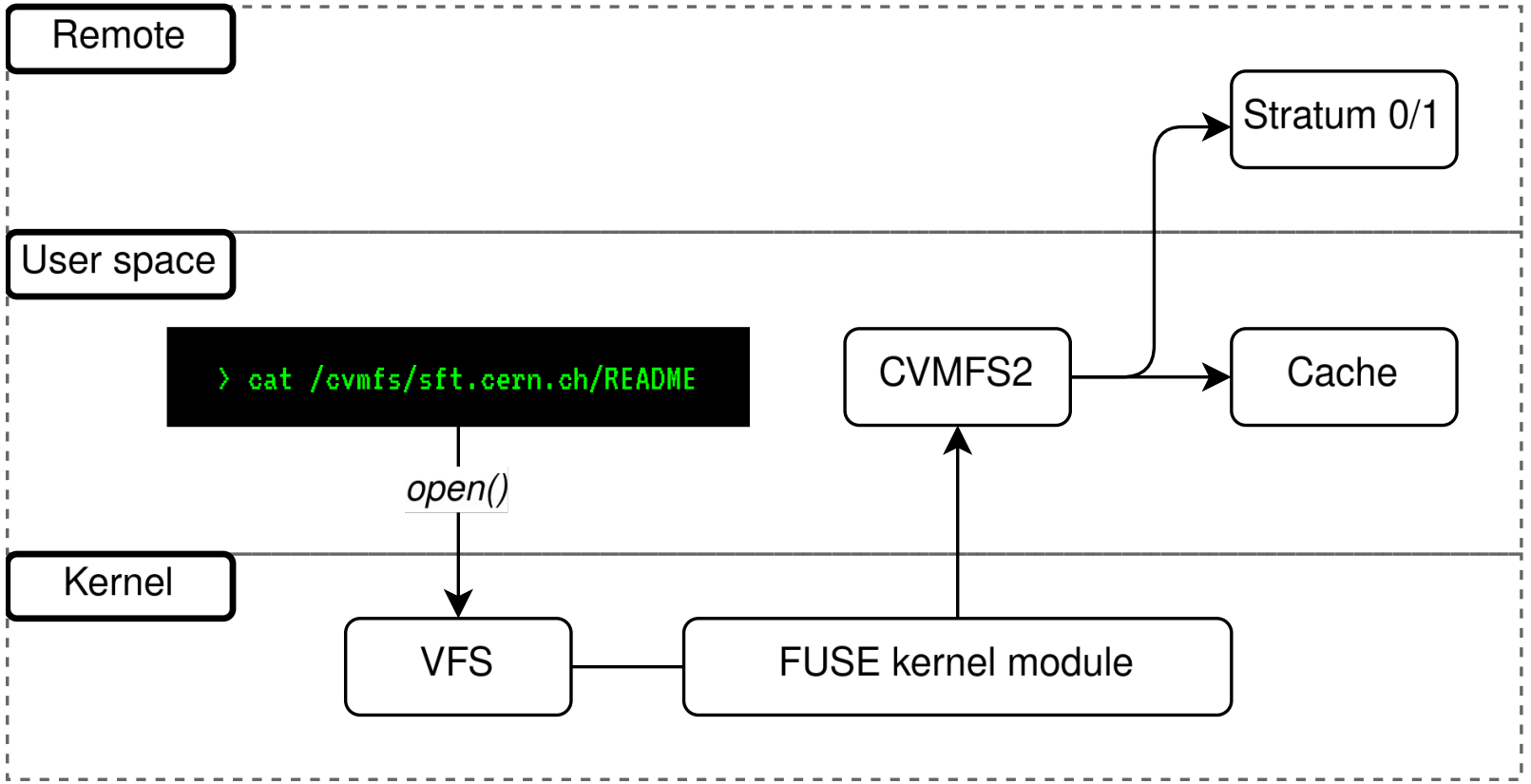
CERN openlab

# What is LxPlus?

- General puprose linux cluster used at CERN
- Over 1000 users daily on average
- Sessions provide access to pre-configured environment
- Provides services like /cvmfs, /eos, /afs
- A big portion of software is distributed using CVMFS

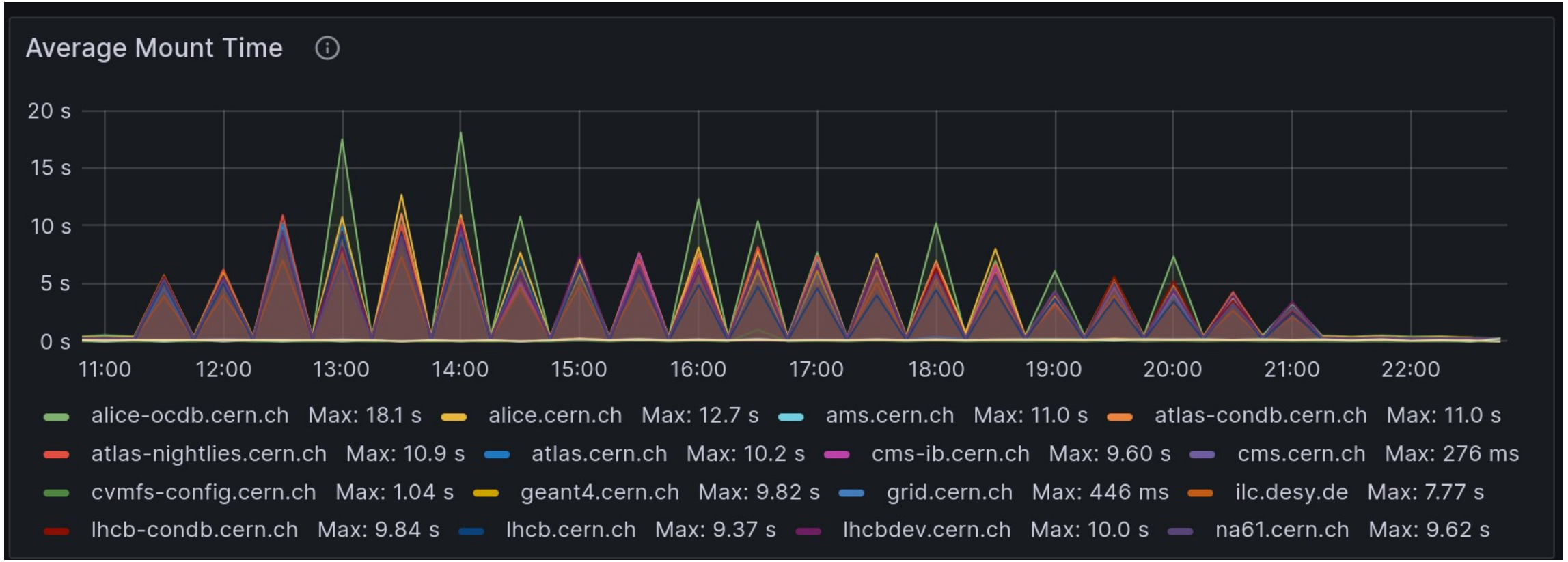
# CVMFS

- Software distribution service
- Read-only filesystem using FUSE
- Widely used on LxPlus



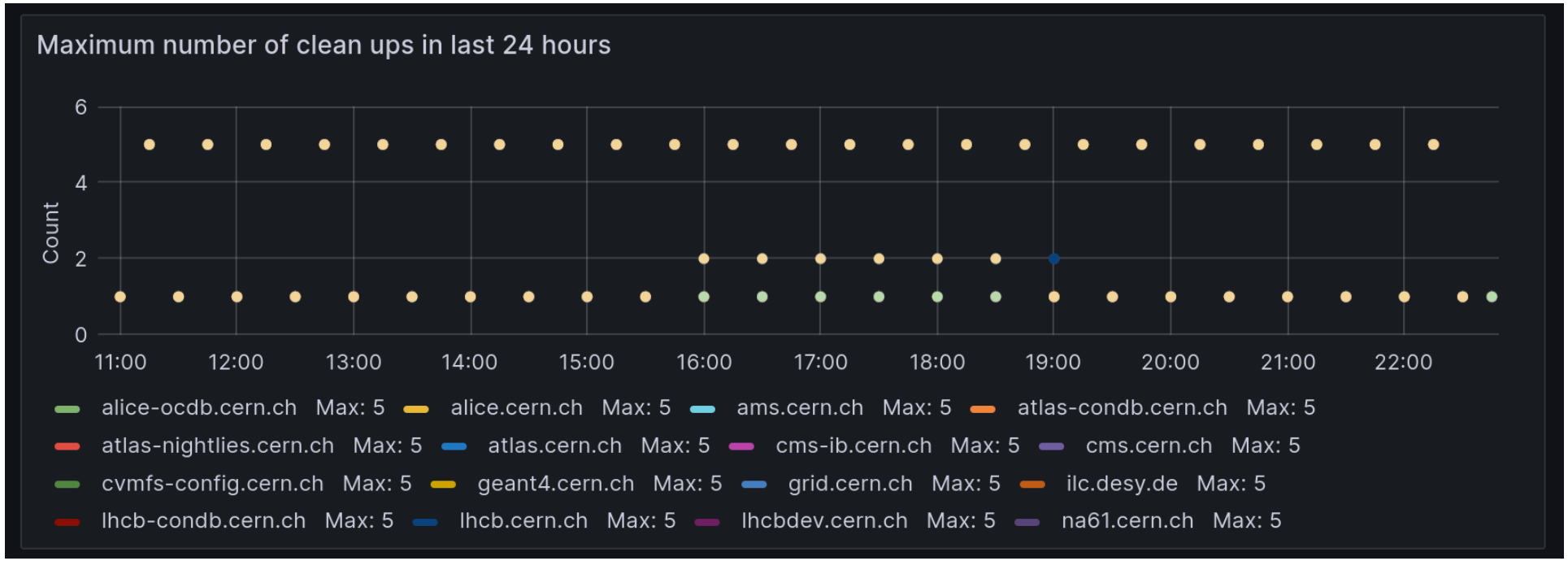
# Current monitoring

# Average mount time



# Max number of cleanups in past 24h

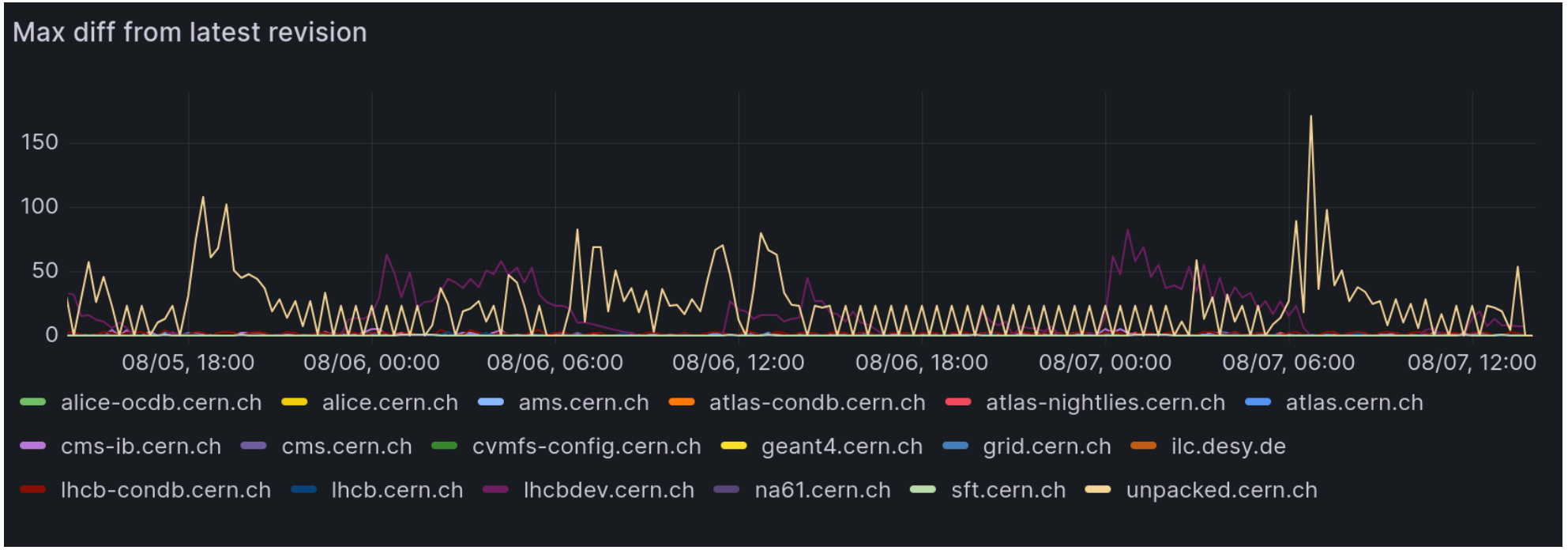
- Alerts when too many (over 20)
  - Usually caused by weird user behaviour, recent example:
    `grep --color=auto -r POG /cvmfs/cms.cern.ch/alma8_aarch64_gcc11 /cvmfs/cms.cern.ch/alma8_amd64_gcc1`



# Data exploration

# Revision monitoring

- Problem: Nodes getting stuck behind
- Solution: Monitor revision numbers and alarm when too far behind
- Uses collectd and Grafana



# Testing hypotheses

- Which repositories take the most space?
- What is the distribution of file sizes?
- What is the distribution of cache sizes?
- How similar are the caches?
- Which files occur most often in different caches?

# Interesting data

**The data:**

- Cache size
- Mounted repositories
- Cache revisions – using extended attributes
- List of files
- File to repository mapping
- File sizes

# List of mounted repos

- We can get it from `mount -l | grep cvmfs`
- Easily parsed using regex “/cvmfs/([^ ]+)”

```
> mount -l | grep cvmfs
/etc/auto.cvmfs on /cvmfs type autofs
(rw,relatime,fd=13,pgrp=1428,timeout=200,minproto=5,maxp
roto=5,indirect,pipe_ino=30848)
cvmfs2 on /cvmfs/lhcb.cern.ch type fuse
(ro,nosuid,nodev,relatime,user_id=0,group_id=0,default_p
ermissions,allow_other)
cvmfs2 on /cvmfs/cms.cern.ch type fuse
(ro,nosuid,nodev,relatime,user_id=0,group_id=0,default_p
ermissions,allow_other)
cvmfs2 on /cvmfs/unpacked.cern.ch type fuse
(ro,nosuid,nodev,relatime,user_id=0,group_id=0,default_p
ermissions,allow_other)
cvmfs2 on /cvmfs/cvmfs-config.cern.ch type fuse
(ro,nosuid,nodev,relatime,user_id=0,group_id=0,default_p
ermissions,allow_other)
```

# Mapping files to repositories

- **Problem:** The cache is shared → no easy way to check which repo a file comes from
- **Naive solution:** Check each repository for each file
- **Better solution:** Memoize which paths we've seen already and map them to repositories as a finite-state machine.

# Mapping sizes to files

- Requires mapping the files to repositories first
- Some files are from unmounted repositories, which requires checking all of them
- If a file couldn’t be found in any repository, it has been deleted in a previous revision

# Measuring root startup

**The example program:**

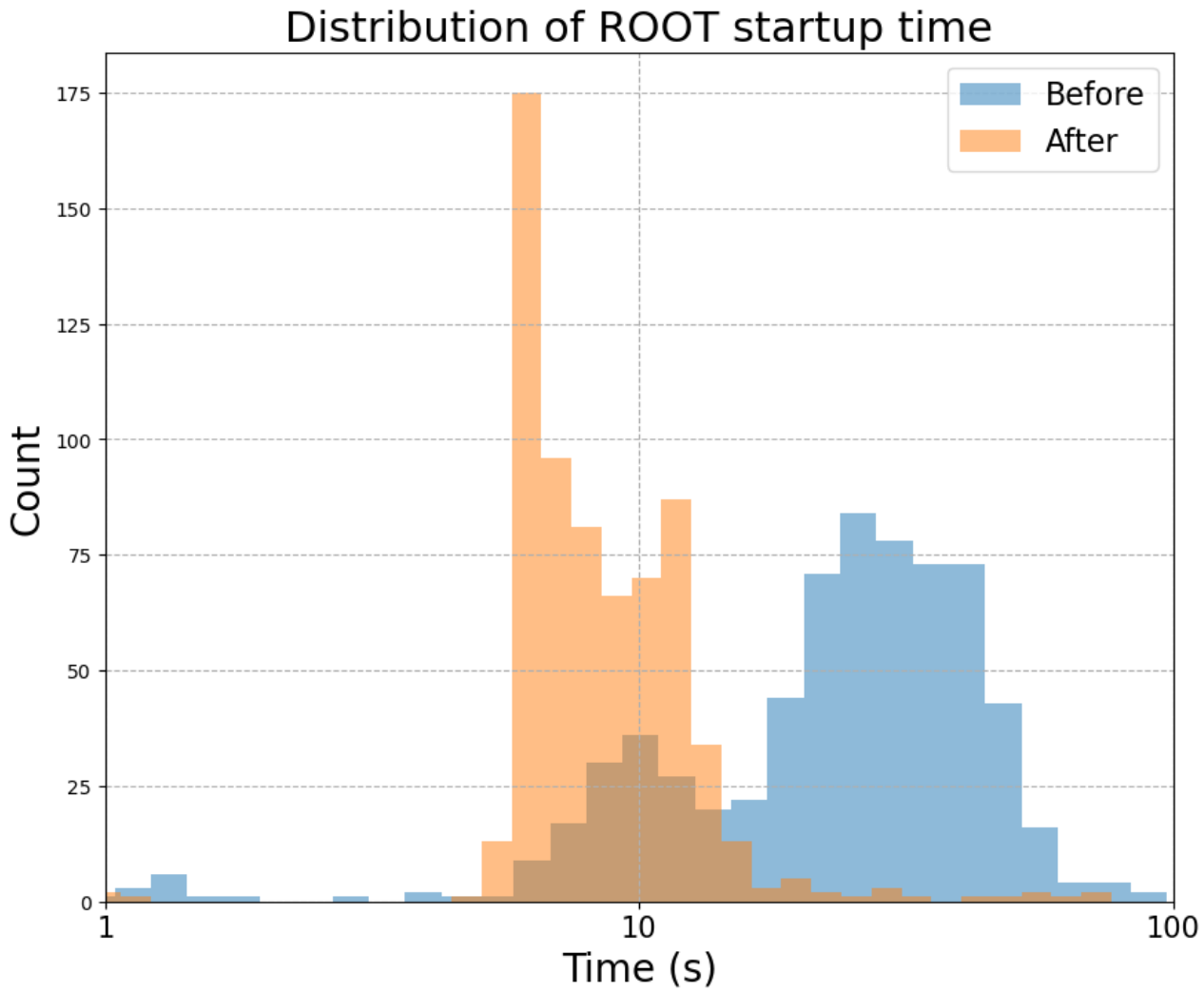
```
root -q -b -e 'ROOT::RDataFrame rdf(100); auto rdf_x =
rdf.Define("x", [](){ return gRandom->Rndm(); }); auto h =
rdf_x.Histo1D("x"); h->DrawClone();'
```

**Method:**

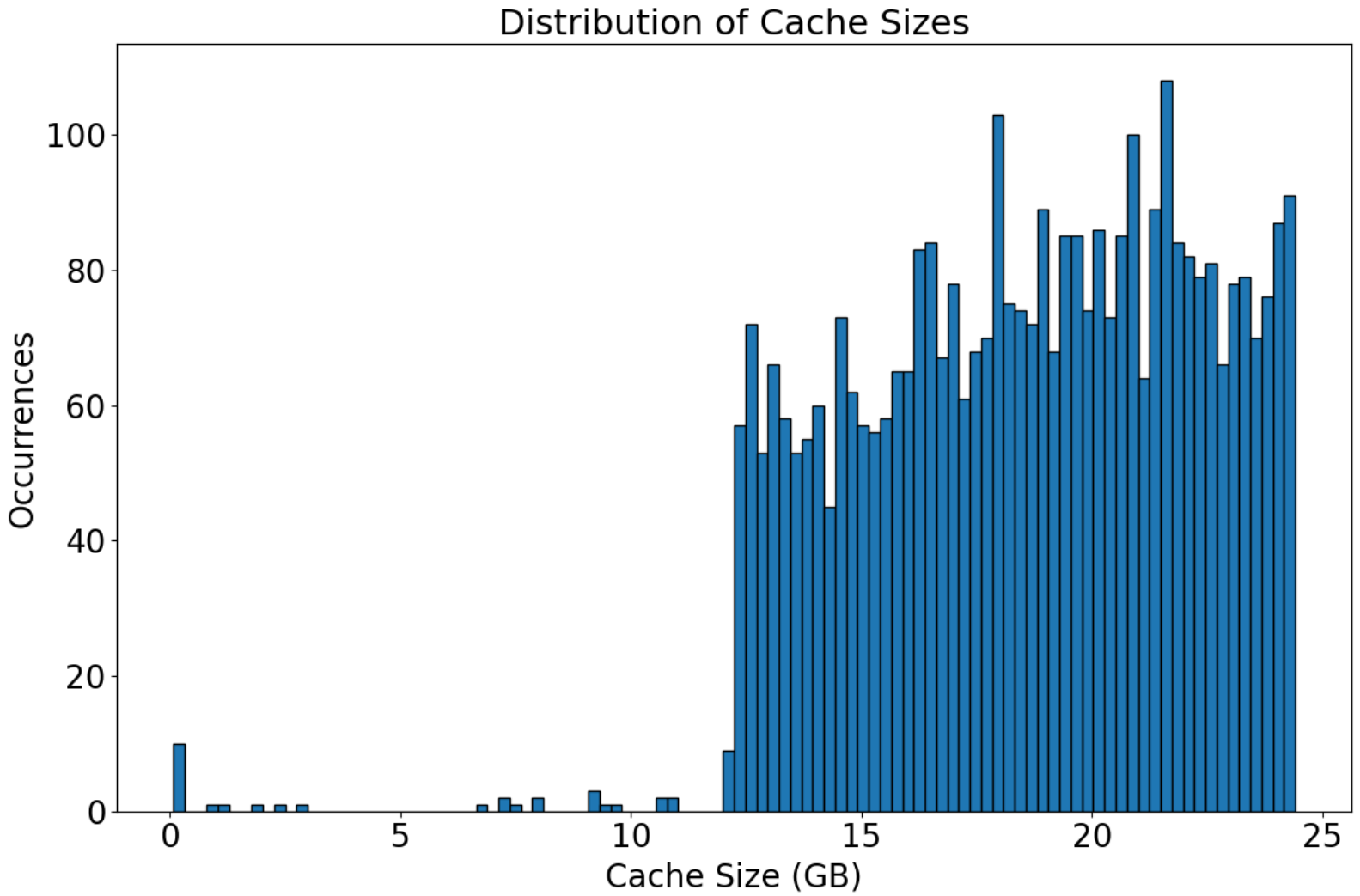
1) Save the cache before running
2) Run root across cluster and measure startup time
3) Save the cache again
4) Run root and measure the time again

# Data Analysis

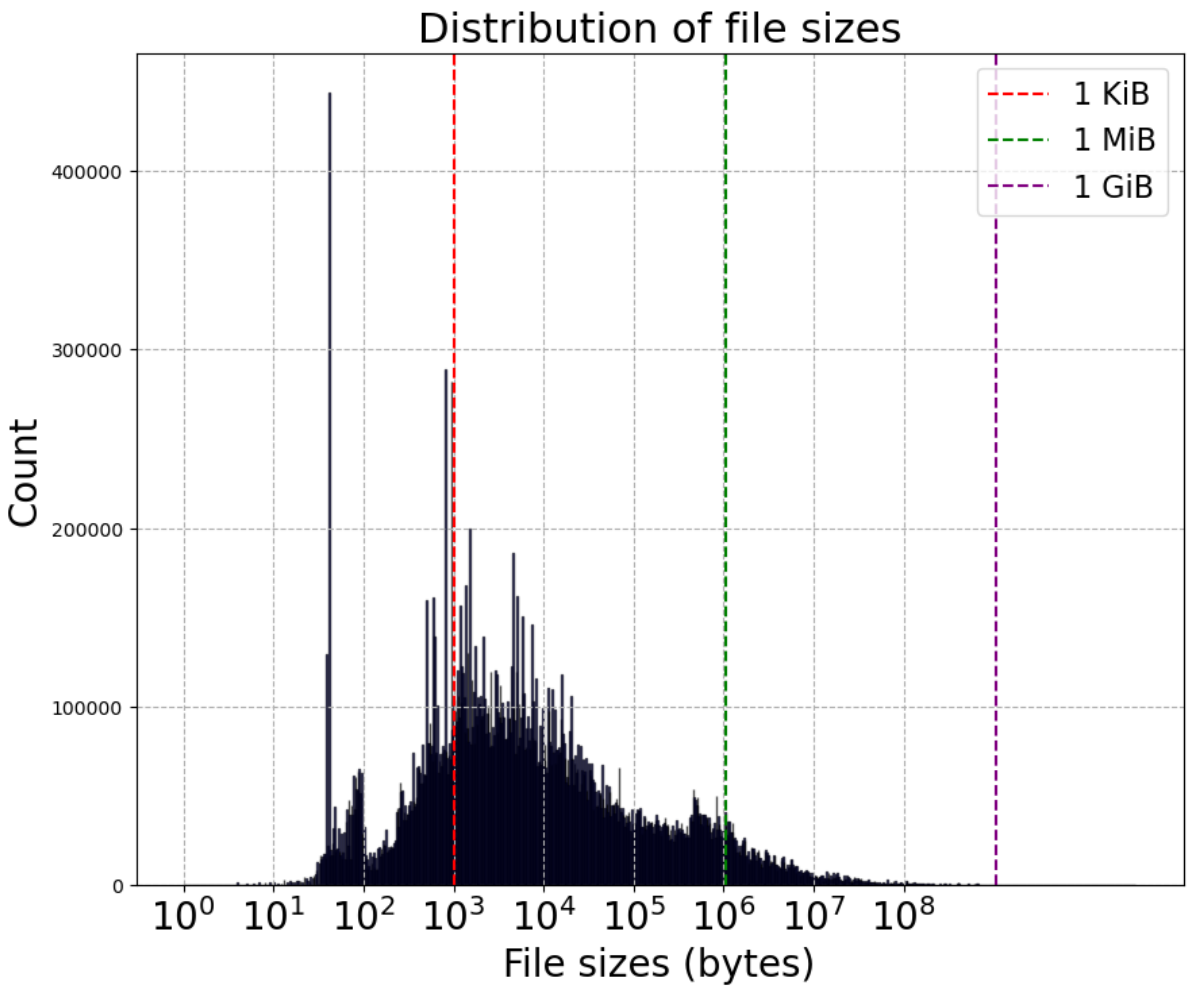
# Root startup time

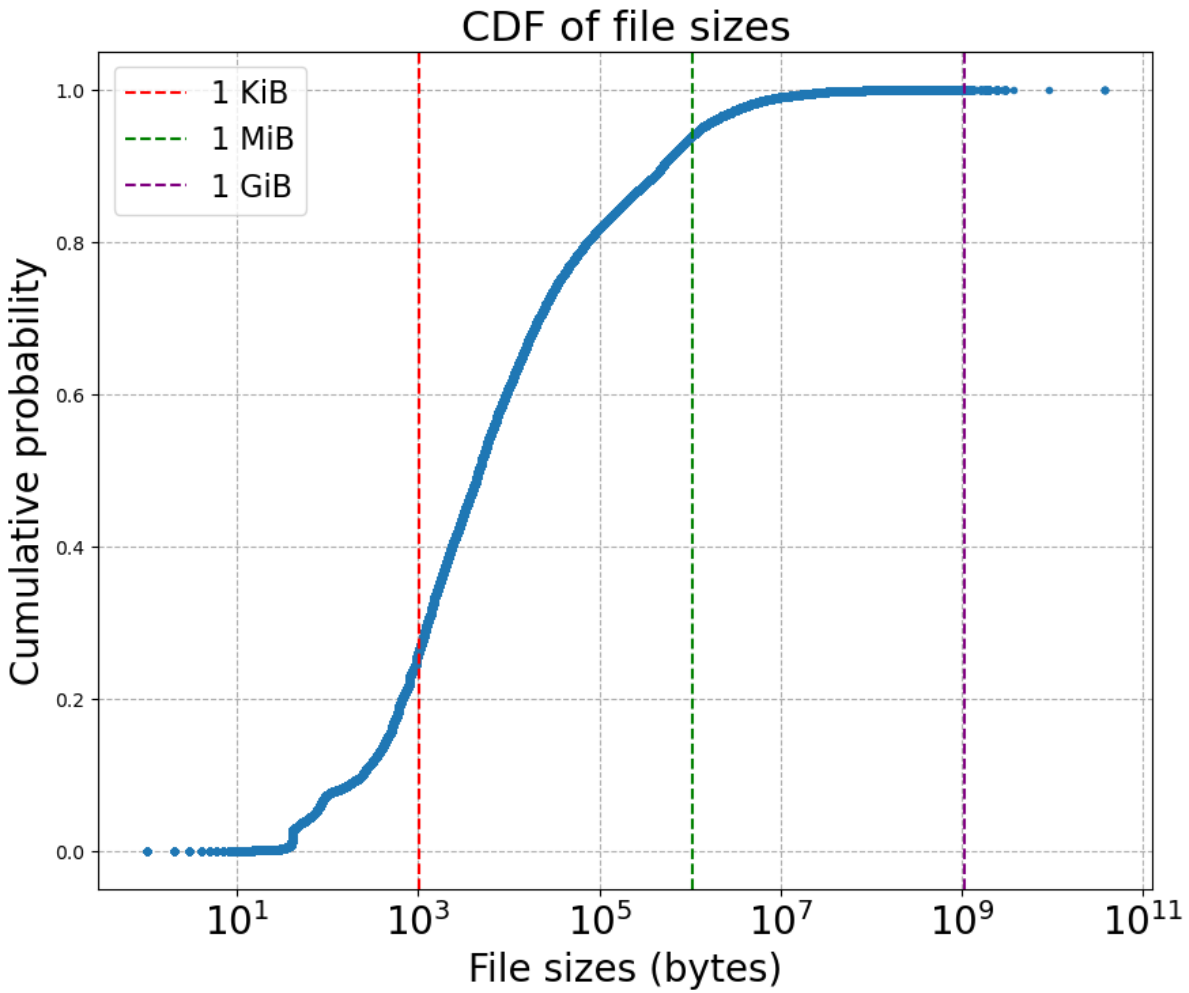


# Cache sizes

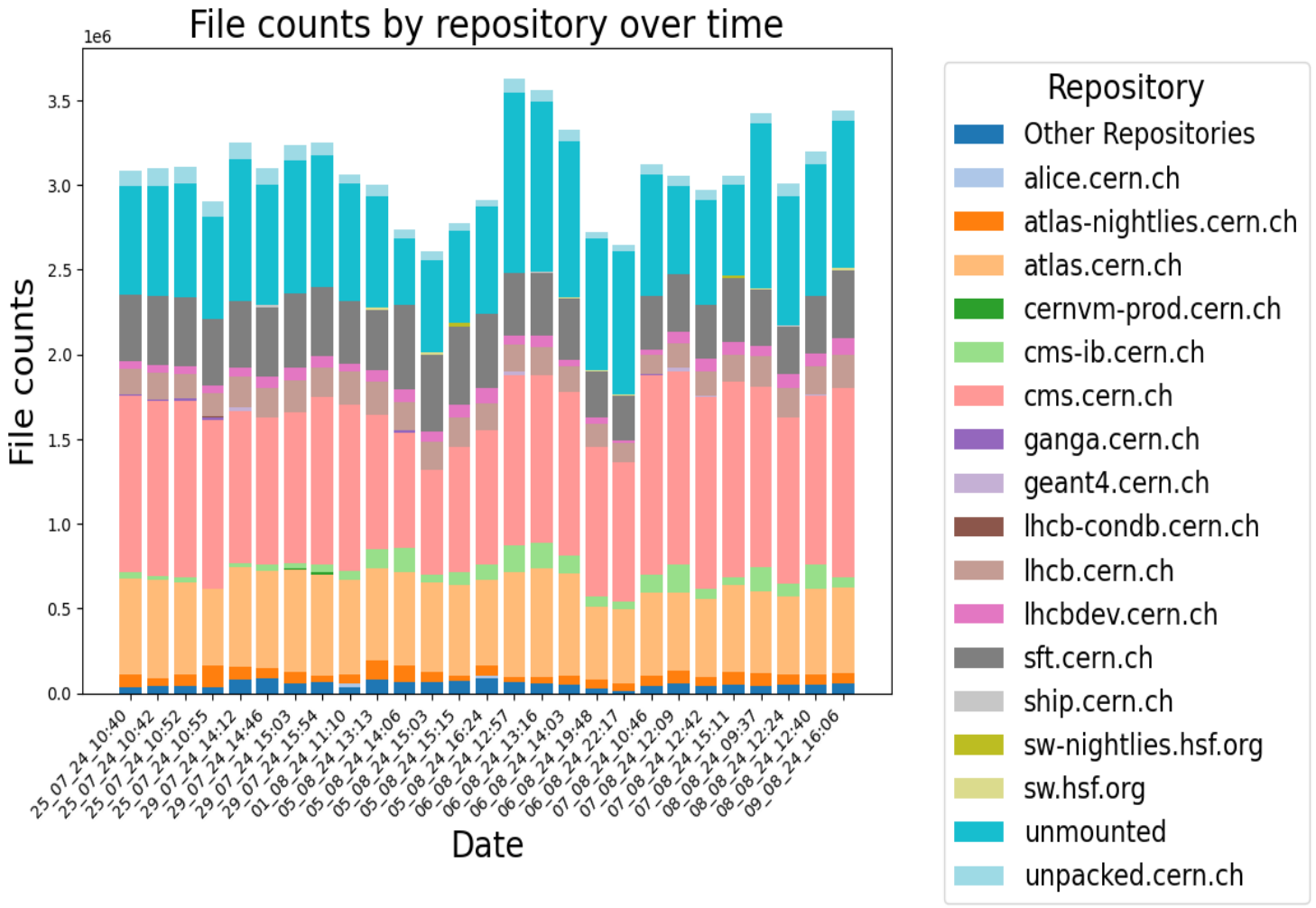

Distribution of Cache Sizes
Occurrences
Cache Size (GB)
0
20
40
60
80
100
0
5
10
15
20
25

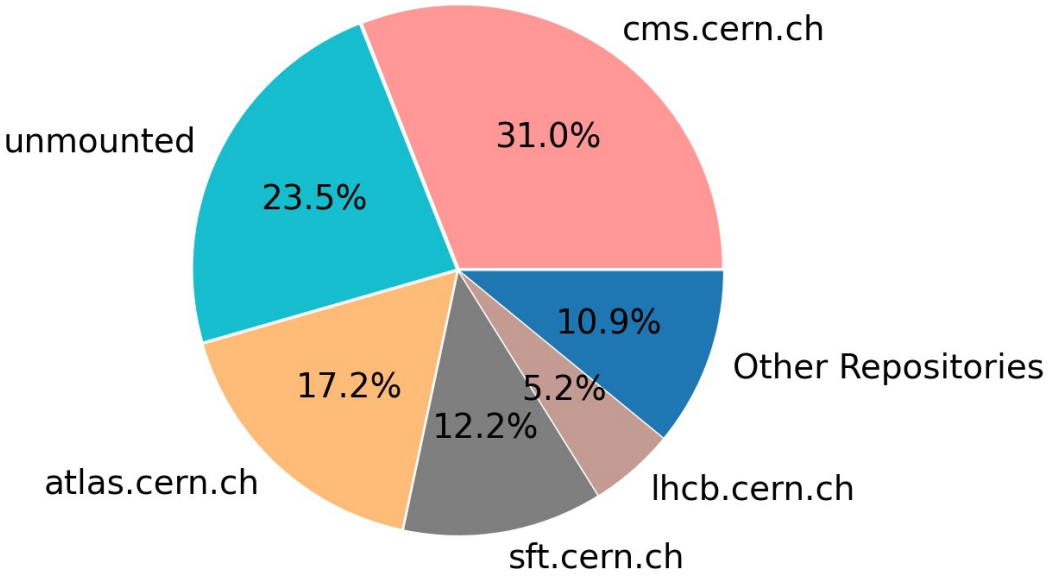
# Distribution of file sizes



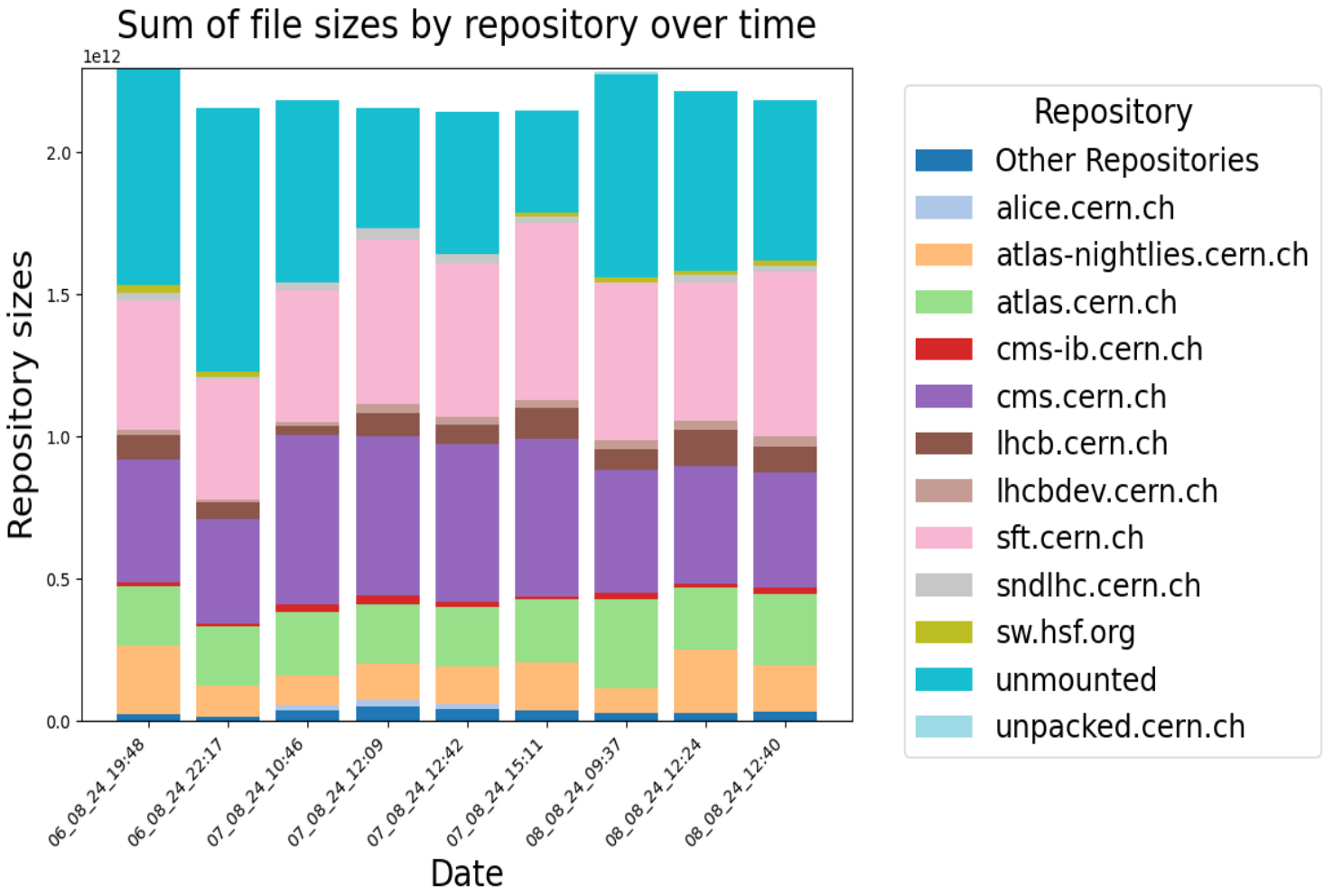



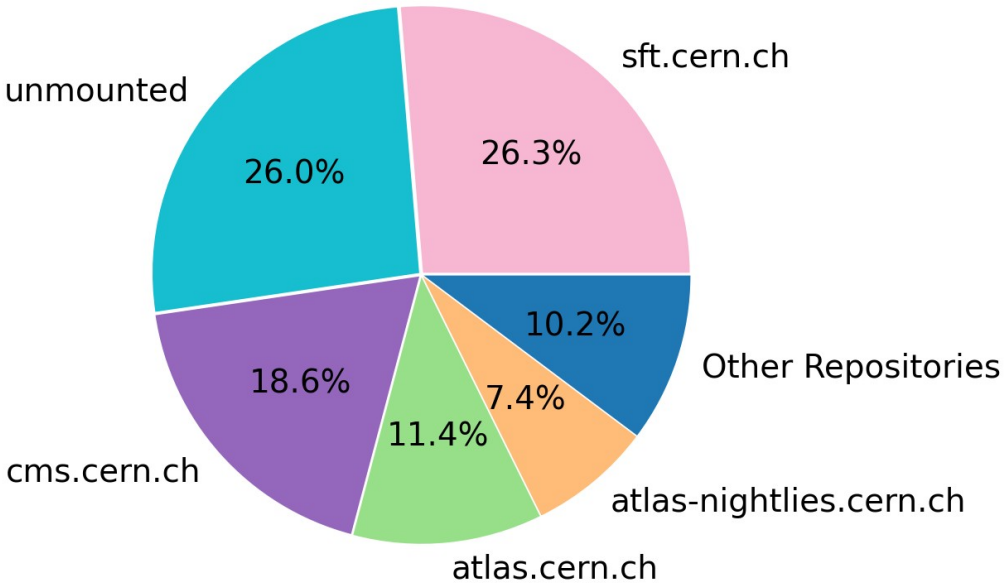
# File counts by repository



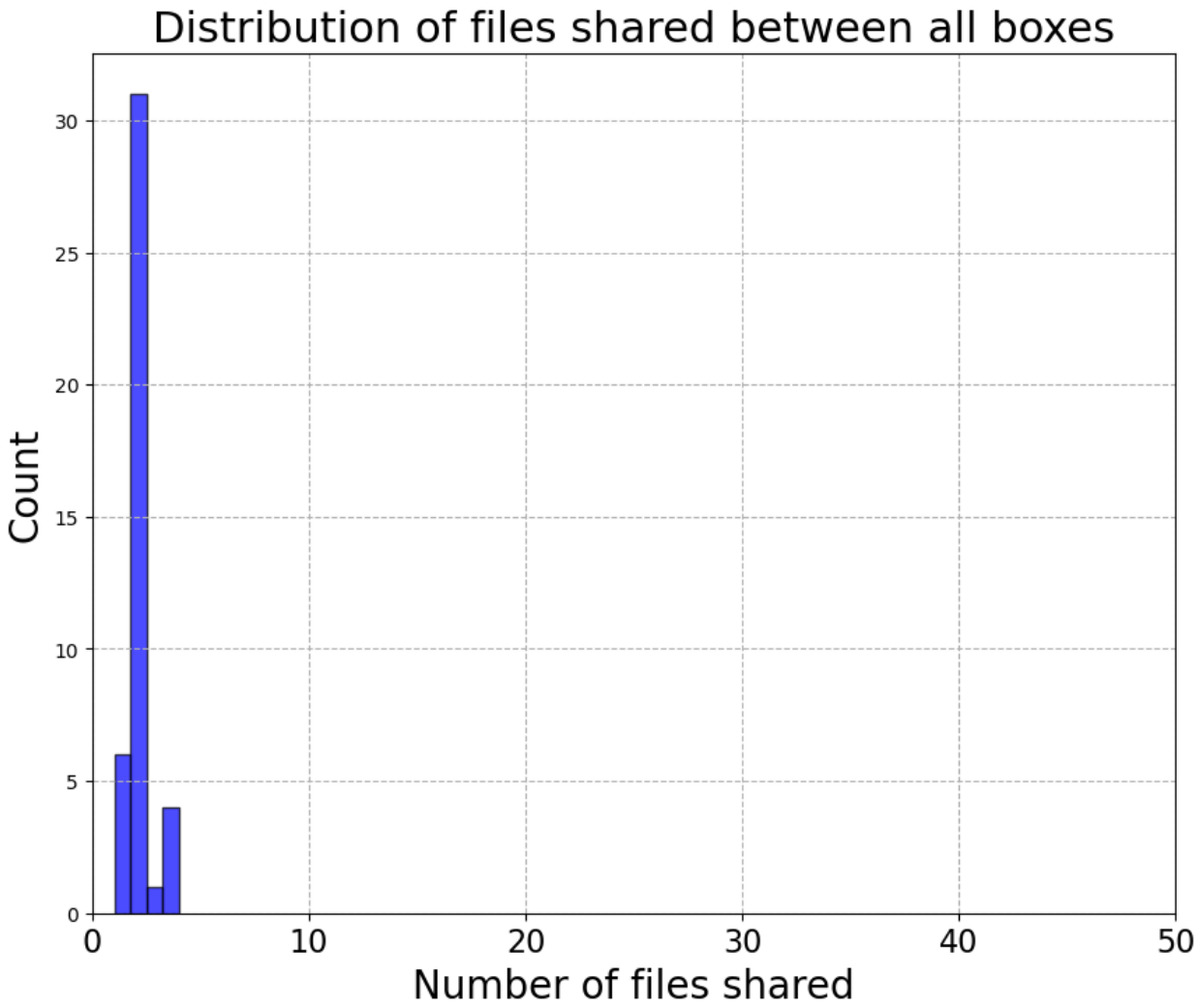



# File sizes by repository

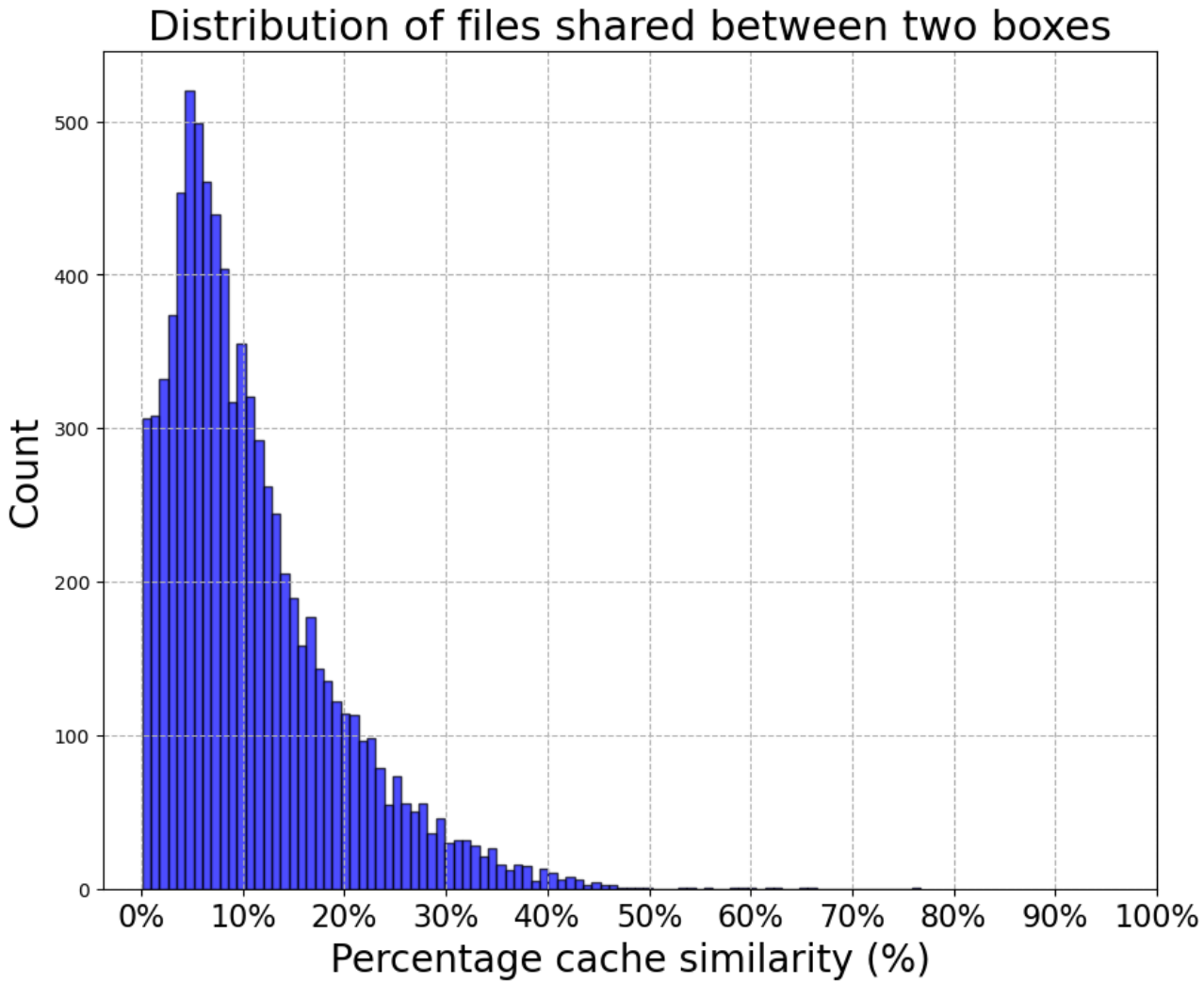


# Cache common elements



# Cache similarity



# Special thanks to

Valentin Volkl

Steve Traylen

# Thank you!

Jakub Ogrodowczyk

CERN openlab