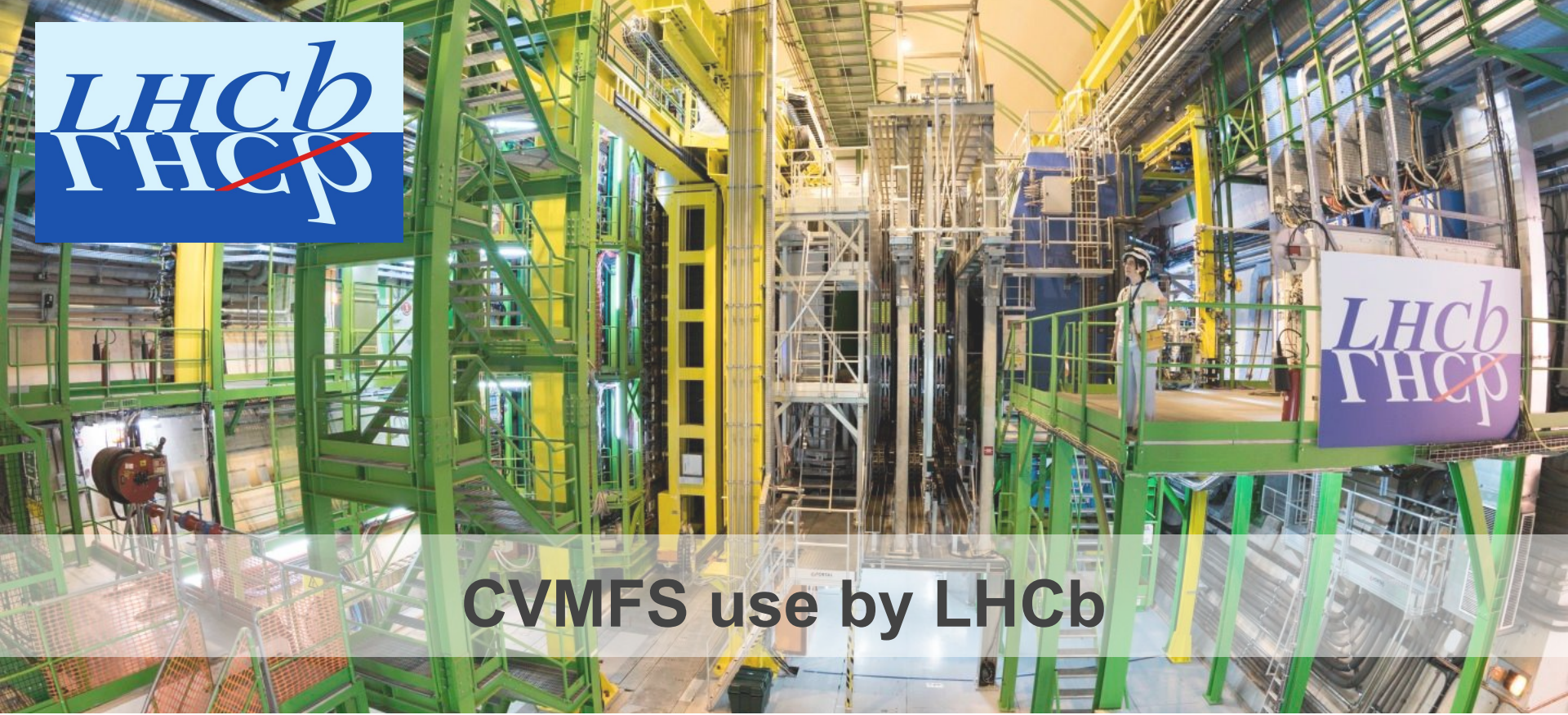


CVMFS use by LHCb

CVMFS User Workshop,
17/09/2024

C.Burr, B.Couturier

CVMFS in LHCb

“CVMFS is the magic thing that means I can get software without thinking about it”

- An LHCb physicist

LHCb CVMFS repositories

LHCb repositories:

- Production software: **/cvmfs/lhcb.cern.ch**
  ⇒ Released software (few changes)
- Conditions: **/cvmfs/lhcb-condb.cern.ch**
  ⇒ Automatic release of conditions (git fetch)
- Nightlies and development software: **/cvmfs/lhcbdev.cern.ch**
  ⇒ continuous integration builds, automated release (very frequent)

They contain all that is needed to process the data in the HLT2 farm, and on the grid

Other repositories are also used e.g. to test the LCG nightlies

What do we distribute on `/cvmfs/lhcb.cern.ch`?

- The “LHCb environment” (aka “LbEnv”)
  - Works on (effectively) any linux machine (x86_64, aarch64, ppc64le)
  - Provides commands to easily setup environments for specific tasks
  - No “source /path/to/file”-ing
- Examples:
  - Start one of the “physics applications”
    - `lb-run AppName/v123`
  - Environments for analysis without LHCb-specific software
    - `lb-conda default[/date]`
  - Grid access
    - `lb-dirac`

What do we distribute on `/cvmfs/lhcb.cern.ch`?

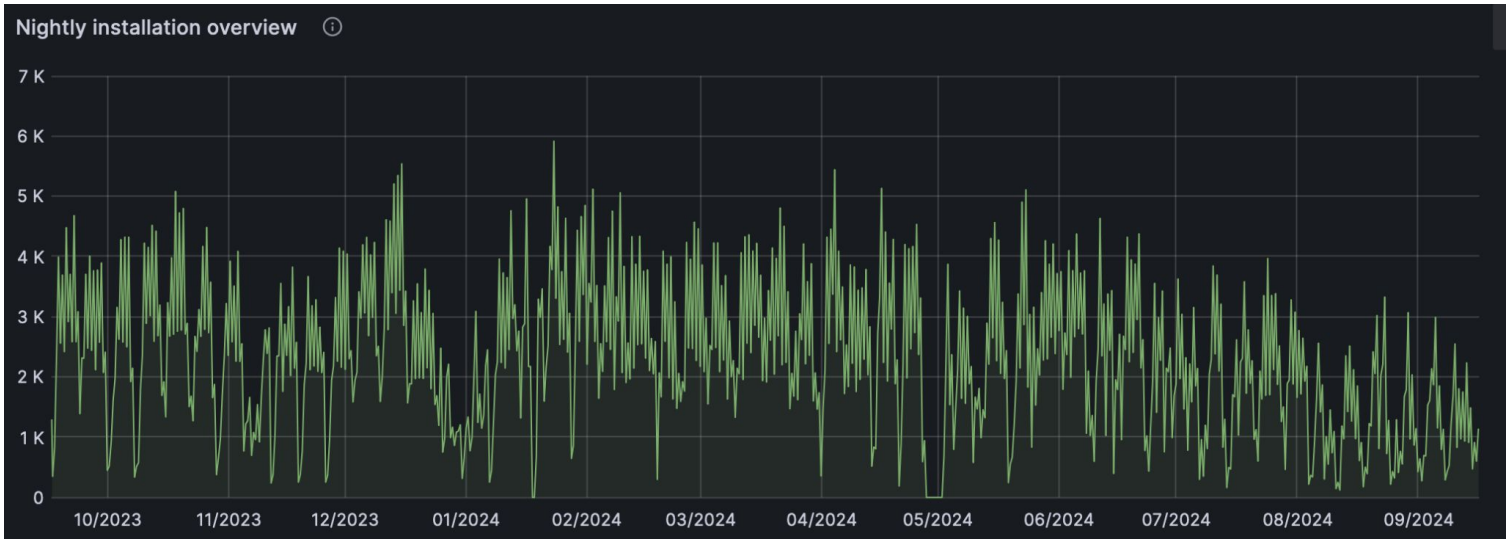
- **ALL** the LHCb production software,

  both latest and legacy versions (since start of LHC data taking)
- Used by all LHCb grid jobs >100k+ at any time
- Standard way to run the LHCb software

  What can we do when CVMFS is not available (e.g. HPCs)?
- **Crucial for long term preservation**

And on `/cvmfs/lhcbdev.cern.ch`?

- Nightly builds
- Intense activity (>1000 packages installed per day)
- One gateway, 6 publishers
- Increased load from “new nightlies” is still a future plan but development stalled



And on `/cvmfs/lhcb-condb.cern.ch`?

- Clones of various git repositories
- Most notably, detector conditions (“GitCondDB”)
- Split from `/cvmfs/lhcb.cern.ch` so we can run cvmfs garbage collection
  - If we did it now there would probably be no need to split

Containers

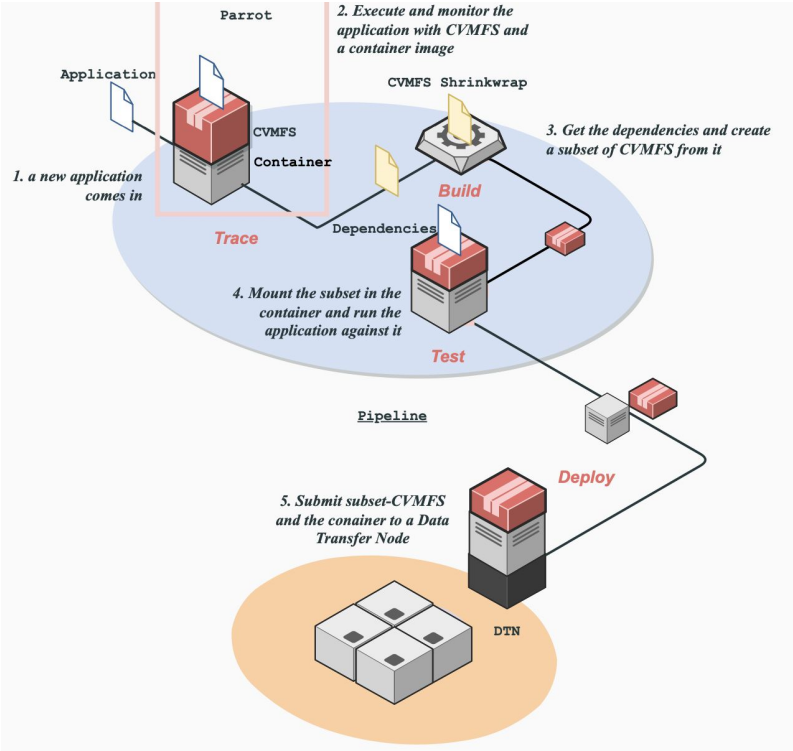
- Previously singularity with `/cvmfs/cernvm-prod.cern.ch/cvm{3,4}/`
- Now we use apptainer (distributed via CVMFS, user namespaces-only)
- Unpacked images in `/cvmfs/lhcb.cern.ch/containers`
  - Would recommend others use unpacked
  - Small reliability improvement from using fewer CVMFS repositories
- `lb-run` launches containers automatically when needed

Multiple architectures

- Most of software installation is architecture independent
- For the couple of exceptions: emulation on the publisher
  - QEMU + binfmt_misc to run a non-native container with apptainer

HPCs

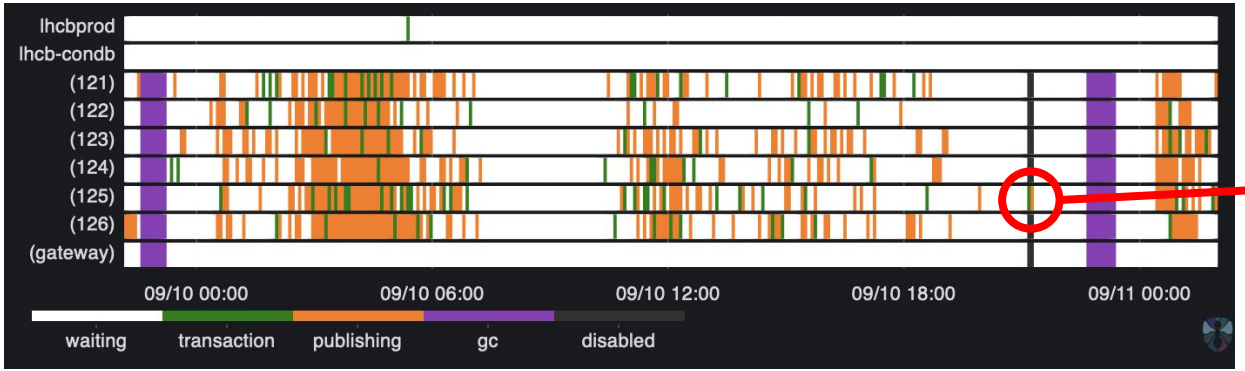
- For HPCs without CVMFS we use “SubCVMFS”
- In summary, copy a subset of CVFMS on to the HPC’s local filesystem
- Only include what is needed to generate current simulation
- 5.4TB lhcb.cern.ch becomes ~20GB



See: https://doi.org/10.1007/978-3-031-07312-0_18

How do things get on to CVMFS?

- Mostly automated: Custom celery application (“lhcb-core-tasks”, see 2021 talk)
  - “Cron”-style period tasks
  - Webhooks from GitLab and Jenkins
- For manual interventions
  - Wrapper shell script around `cvmfs_server transaction/publish`
  - Needed to pause the automatic activity before starting a transaction



Manual transaction stopped other lhcbdev nodes

Nitpicks

- Client reliability issues at scale
  - Often need to open tickets about CVMFS issues on grid nodes
  - Occasional nightly build farm issues
  - Medium operational burden

⇒ *improving communication with the sites could help ?*

- Client performance with cold cache
  - “Sometimes I have to wait 30 minutes to start ROOT” - PhD student
  - See also [cvmfs/cvmfs#3078]

Word cloud of LHCb GGUS ticket subjects
→ flawed representation but it’s pretty..

Conclusion

THANKS !

- We’re very happy with CVMFS itself
- We’re also very happy with the quality of service provided by CERN IT
- Our deployment system has been very low maintenance for years
- Reliability and performance improvements would be welcome in some situations