



# Analysis Portability (aka software containers)

Doug Benjamin (Brookhaven National Lab)

Oksana Shadura (University of Nebraska, Lincoln)

6-May-2023



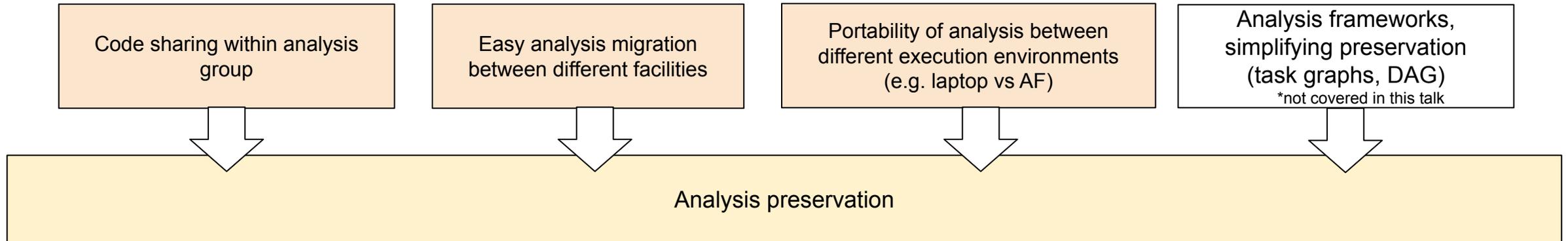
@BrookhavenLab

# Acknowledgements

Maria Acosta Flechas, Jakob Blomer, Lincoln Bryant, Matthew Feickert, Rob Gardner, Lindsey Gray, Alex Held, Fengping Hu, Ofer Rind, Ricardo Rocha, Heidi Schellman, Oksana Shadura, Valentin Volki, Ilija Vukotic, Wei Yang, Shuwei Ye

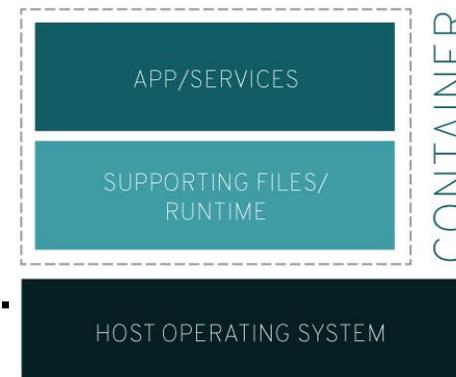
# Analysis portability from user interface point of view

- **Portability within the same Analysis Facility** (e.g. sharing code within analysis group)
  - Analysis should be easily portable between two different members from the same group or other member of collaboration
- **Code migration between different Analysis Facilities**
  - Analysis, metadata and data in addition should be easily portable between different Analysis Facilities
- In addition the same analysis (with minimal boilerplates) should be easily executed locally on user laptop



# In the beginning (ie Run 1) ... CVMFS

- There was CVMFS.
  - Prior to start of Run 1 – ATLAS decided to use cvmfs for code distribution – started in the US ATLAS Tier 3 program.
  - Very beneficial for having a consistent UI across the grid, CERN, local Tier 3 sites.
- 2008 – Docker came to be.
  - Users can now create their own software containers for their analysis environment
- CVMFS can be used to deploy containers. (/cvmfs/unpacked.cern.ch)
  - Used extensively across the grid



# /cvmfs/unpacked.cern.ch/...



- Open to all CERN users. [Link to CVMFS Documenation](#)
  - List of images – JSON file - [Link JSON file in CERN Gitlab](#)
  - Images added by merge request/ pull request.
    - Pull Requests manual vetted by a group of people at CERN
    - Once on the list, system automatically picks up changes and new tags (if wildcards are used for tags in the images specification)
- Image synchronization runs every 20 minutes and available to CERN and outside CERN within 45 minutes.
- Designed to handle users creating a new image daily
- The unpacked service benefits for proper image layering
  - Best Practice if users use analysis framework base image – change only the layer with their analysis code.

# CERN Harbor Registry



HARBOR

- CERN Harbor Registry – allows users to store and secure user images, helm charts or OCI artifact.
  - Documentation - <https://kubernetes.docs.cern.ch/docs/registry/>
    - Including a [Quick start Guide](#)
  - All CERN users can create a new project and repository default quota 20 GB and can request a significant increase.
  - Create [Service Desk Ticket](#) to create new project or increase quota
  - Various tools (scans, immutability and garbage collection, etc) available directly to users.
- Avoids the issues with changing DockerHub policies
- Common place to share/exchange containers and best practices

# Analysis containers in ATLAS

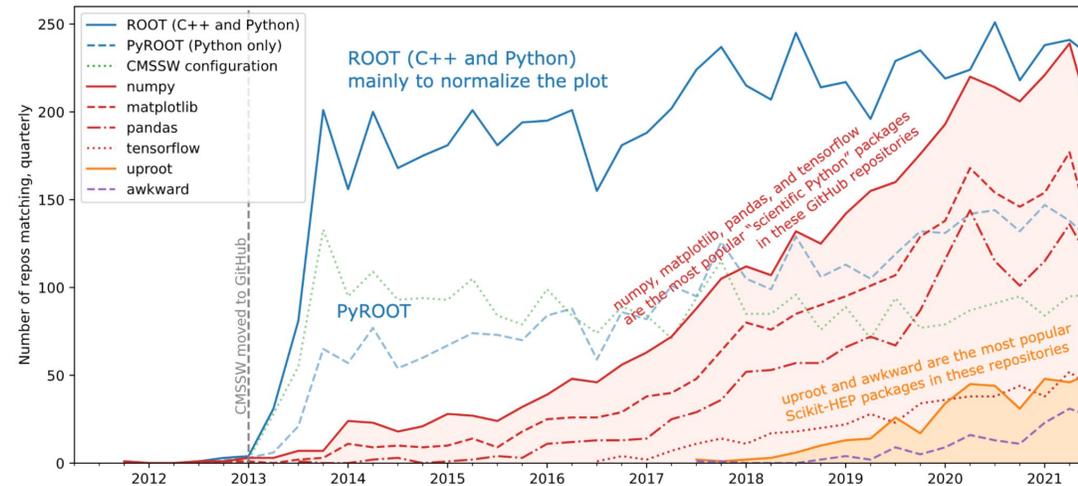
- [AnalysisBase](#) and [AthAnalysis](#) are available CERN gitlab container registry and also published through `/cvmfs/unpacked.cern.ch`
  - ATLAS has been able to run custom containers on the grid for ~ 5 years.
- New effort in the ATLAS Analysis Model group (AMG) to harmonize and produce other types analysis containers
  - US Analysis Facilities at (BNL, SLAC, Univ. of Chicago) building ML container to provide the same user environment and tools. Each AF has significant enough differences to preclude one size fits all strategy.

# CMSSW containerization efforts

- CMSSW official images image - <https://github.com/cms-sw/cms-docker>
  - Available in **hub.docker.com** and at cvmfs through **unpacked.cern.ch** (/cvmfs/unpacked.cern.ch/registry.hub.docker.com/cmssw)
  - Multiple architectures and support of operations systems:
    - [aarch64, x86\_64, ppc64le] and [alma8, cc7, cc8, cs8, cs9, el6, el7, el8, rhel7, rhel6, rhel7, rhel8 and others]
- Alternatively CMSSW images maintained by CMS Data Preservation and Open Access Group - <https://gitlab.cern.ch/cms-cloud/cmssw-docker>
  - Available in **gitlab-registry.cern.ch** and at cvmfs through **unpacked.cern.ch** (/cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/cms-cloud/cmssw-docker)
  - Used in REANA to ensure Open Data analysis examples are preserved

# End-user CMS analysis framework containerization

- RDF / PyROOT
  - Available as a part of CMSSW containerisation effort (through packaged in CMSSW ROOT)
- Coffea
  - *coffeateam/coffea-base* and *coffeateam/coffea-dask* images available [hub.docker.com](https://hub.docker.com) and at cvmfs through [unpacked.cern.ch](https://unpacked.cern.ch)



"Status of Analysis - The Python Perspective", Jim Pivarski, [Future Trends in Nuclear Physics Computing](#)

# What about Data?

- Analysis Portability is more than software environment.
- Users need input data and often need to share their output with others
- Tools for Sharing data/Collaboration
  - Standard experiment methods (Rucio/fts, xrootd, cvmfs), wiki pages, gitlab repos, EOS, google drive
- US Analysis facilities (BNL, SLAC, Univ of Chicago) all fuse mount CERN EOS instance on their interactive resources
  - R/W access granted through CERN Kerberos credential
  - Do not know the scaling limitations of this approach - but users like Posix
- Should CERN Box be used as common sync/share storage (uniform “home” area for all users)

# Portability of end-user-analysis between collaborators

- Encourage users to **follow best python development practices** for build system and packaging (as well for C++ code)
- **Maintain virtualenvs / conda environment and/or update requirements.txt**
  - Virtualenvs (venv) are still “pain point” in case *venv+cmssw* or *venv+LCG* (both of these not properly setting system python path to their local area)
  - It works well in the case *venv+coffea-dask image*
- Maintain in analysis repositories CI pipelines (GH Actions / Gitlab) with at least (!) minimal set of tests (applicable as well for C++ code)
  - Helps to catch issues with different python / c++ versions
  - Help to keep up-to-date analysis package dependencies
  - Avoid hardcoded path and etc.
  - Allow to add easily other useful checks (e.g. lint)

# Portability of end-user analysis between CMS facilities

- Current status at CMS AFs:
  - **CMS AF @ UNL:** coffeateam/coffea-casa-\* images (based on coffeateam/coffea-dask images)
  - **EAF FNAL:** LPC sl7 & coffea-dask-sl7 images (based on coffeateam/coffea-dask images) + GPU flavours
  - **CMS AF @ Purdue:** geddes-registry.rcac.purdue.edu/cms/cmsaf-alma8
  - **NAF @ Desy:** cmssw/slc6 & cmssw/slc7
  - **INFN AF:** dask-cc7 & dask-alma8 (based on cern/alma8-base & cern/cc7-base)
- Encourage analysis groups to test analysis docker container
  - Test maintained baseline containers provided by CMS collaboration / analysis groups
- Encourage facility maintainers to add support for Binderhub **to share reproducible interactive computing environments from code repositories** (primary technology behind <https://mybinder.org/>)

# Portability of analysis between different execution environments (e.g. laptop vs AF)

- For now no centrally supported effort
  - *Coffea* has various executors allowing to put it on any compute substrate and *RDF* provides *distRDF library* with *dask* and *spark*
  - Investigate libraries such *ray*, *parsl*, and *taskvine*, that intend to take *dask* task-graphs and apply different opinions on scheduling them
    - Exists an **issue of environment consistency, transporting metadata (like corrections)**, etc.
    - Partially could be resolvable by containers
- Some of analysis have **boilerplated code** (see *runner.py*) allowing to support multiple backends, facilities, as well run analysis locally

```
93
94 # Scale out
95 parser.add_argument(
96     "--executor",
97     choices=[
98         "iterative",
99         "futures",
100         "parsl/slurm",
101         "parsl/condor",
102         "parsl/condor/naf_lite",
103         "dask/condor",
104         "dask/slurm",
105         "dask/lpc",
106         "dask/lxplus",
107         "dask/casa",
108     ],
109     default="futures",
110     help="The type of executor to use (default: %(default)s). Other options can be implemented. "
111     "For example see https://parsl.readthedocs.io/en/stable/userguide/configuring.html"
112     "- 'parsl/slurm' - tested at DESY/Maxwell"
113     "- 'parsl/condor' - tested at DESY, RWTH"
114     "- 'parsl/condor/naf_lite' - tested at DESY"
115     "- 'dask/slurm' - tested at DESY/Maxwell"
116     "- 'dask/condor' - tested at DESY, RWTH"
117     "- 'dask/lpc' - custom lpc/condor setup (due to write access restrictions)"
118     "- 'dask/lxplus' - custom lxplus/condor setup (due to port restrictions)",
119 )
```

[Example of analysis portability between AFs with different scaling option \(\*runner.py\*\)](#)

# Issues to ponder (ie for discussion)

- **How do we handle small user extensions to their software environment?**
  - Is the container building process so quick and painless that you build a new container?
  - Install into site local shared filesystem (ie home area)
- **How do we handle the co-scheduling the variety of workflows and activities (CLI, batch, jupyter notebooks, Service X, Coffea-Casa)?**
  - integrate and operation HTC scheduling systems (HTCondor) and Kubernetes platforms.
- **Can/Should CERNBox be used for a common storage across all analysis resources?**
  - Users like posix interfaces - easy to use. Some common R/W resource is need, accessible from CLI, laptop, or jupyter notebook

# Conclusions

- **Analysis Portability is important to meet analyzers where they are.**
  - Goal - Increase analyzers productivity
  - Requirements include -
    - Code sharing with analysis group
    - Easy analysis migration between different facilities
    - Analysis portability between analysis environments (laptop vs analysis facility)
- **CVMFS has proven to be a very valuable tool and continues to be a vital**
- **Software containers are a piece of the puzzle for Analysis portability**
- **Data access and Data storage/sharing are an important aspect.**
  - Several ideas but no common solution has been adopted.

# References

<https://cvmfs.readthedocs.io/en/stable/index.html>

<https://www.redhat.com/en/topics/containers/whats-a-linux-container>

<https://www.redhat.com/en/topics/containers/containers-vs-vms>

<https://www.redhat.com/en/topics/containers/what-is-container-orchestration>

[https://cern.service-now.com/service-portal?id=service\\_element&name=kubernetes-service\\_is](https://cern.service-now.com/service-portal?id=service_element&name=kubernetes-service_is)

<https://kubernetes.docs.cern.ch/docs/registry/>

<https://cvmfs.readthedocs.io/en/stable/cpt-containers.html> - container-images-and-cernvm-fs

<https://cvmfs.readthedocs.io/en/stable/cpt-containers.html#distributing-container-images-on-cern-vm-fs>

<https://gitlab.cern.ch/unpacked/sync/-/blob/master/recipe.yaml>