

Analysis Facility Introduction: Users requirements

WLCG-HSF Pre-CHEP Workshop 2023

[Whitepaper](#)
[Session notes to contribute to](#)



Introduction

The LHC: successful history of distributed computing infrastructure

- globally distributed resources with equitable access to all collaborators

But also:

- *We have ~15 years of data-taking experience & HL-LHC looms large*
- *Computing and data science is developing fast*

How will/should/can our analysis infrastructure change?

HSF Analysis Facility Forum (Whitepaper)

Started in March 2022 with Analysis Facility Kick-off [\[indico\]](#)

- roughly monthly meetings [last: April '23]

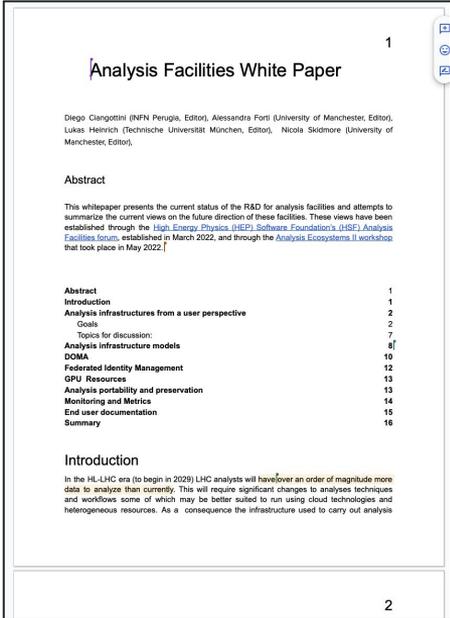
Mandate:

- Provide Forum to discuss efforts across the community
- Collect main ideas in a Whitepaper
 - Drafted by coordinators to provide basis for discussion
 - Goal: collect broad community views → HSF authorship/endorsement

[HSF AE II Workshop report](#)

Coordinators:

A. Forti & L. Heinrich (ATLAS), N. Skidmore (LHCb), D. Ciangottini (CMS)



1

Analysis Facilities White Paper

Diego Ciangottini (INFN Perugia, Editor), Alessandra Forti (University of Manchester, Editor),
Lukas Heinrich (Technische Universität München, Editor), Nicola Skidmore (University of
Manchester, Editor).

Abstract

This whitepaper presents the current status of the RAD for analysis facilities and attempts to summarize the current views on the future direction of these facilities. These views have been established through the [High Energy Physics \(HEP\) Software Foundation's \(HSF\) Analysis Facilities forum](#), established in March 2022, and through the [Analysis Ecosystems II workshop](#) that took place in May 2022.

Abstract	1
Introduction	1
Analysis Infrastructures from a user perspective	2
Goals	2
Topics for discussion:	7
Analysis infrastructure models	8
DCMA	10
Federated Identity Management	12
GPU Resources	13
Analysis portability and preservation	13
Monitoring and Metrics	14
End user documentation	15
Summary	16

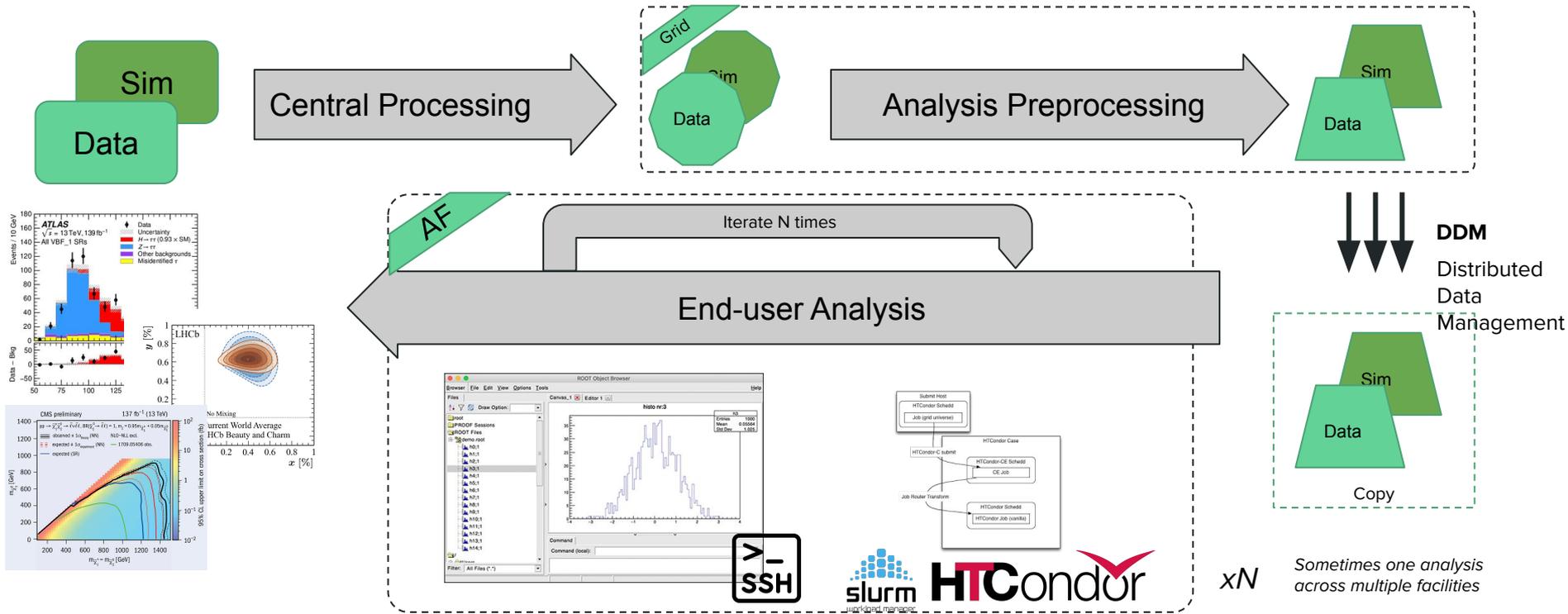
Introduction

In the HL-LHC era (to begin in 2026) LHC analysts will have over an order of magnitude more data to analyze than currently. This will require significant changes to analysis techniques and workflows some of which may be better suited to run using cloud technologies and heterogeneous resources. As a consequence the infrastructure used to carry out analysis

2

The Starting Point & Observations

A classic example analysis workflow at large Experiments on existing AFs



The Starting Point & Observations

- This mode has served us well and is supporting analyses now.
 - “*Analysis Facilities*” not a new idea
- Analyzers often **leave the global Grid** asap to work on **local resources**
 - Analysis Resources “darker” from a VO point of view
 - Advantages analyzers at big institutions
- **“Interactive Analysis”** is mostly defined by what fits onto a **single machine** (plotting, fitting, playing with cuts, ...) - rest must go to batch (→ latency)
 - Constrained to e.g. GB scale analysis vs. TB
 - Large-scale “end-to-end” interactive analysis not a focus of analysis experience today

The Starting Point & Observations

- **Authorization & Authentication (AAI)** is different across global and local resources
- Analysts do **manual data preparation & management**
 - Preparation of custom formats, skimming events, /... on grid
 - Often aware of data details (e.g. data locality)
 - Changes in upstream processing can take high-latency iterations
- **Data Artifact Sharing** more via local filesystems than via DDM
 - Once you go to the AF you often don't go back to the Grid
- **Analysis is evolving** beyond “classic compute slots”
 - e.g. Machine Learning has its own culture, practices & needs

Access to Facilities

Large-Scale HENP* experiments are *a global enterprise*:

- Key: equitable access to a global pool of resources for any VO member.
- Some consequences in case of lacking access::
 - Analysis team is scattered across multiple facilities (complex code, duplicate data)
 - Advantages people with bigger institutions / regions / countries
 - Those w/o options flock to the one facility where everyone has an account: CERN/Hostlab

Integration into federated infrastructure

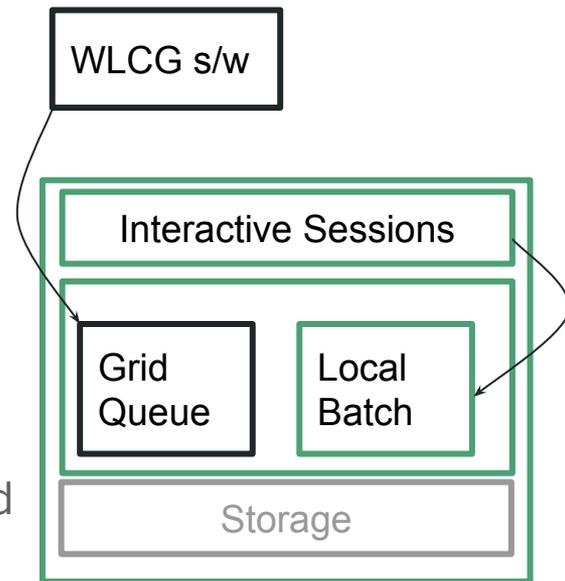
From current analysis facilities it's possible to interact with the global infrastructure

- send jobs, receive data
- if grid-site co-located: also receive jobs
- at the same time AF currently are very distinct from grid

Frequently stated: facility should be able to do full analysis lifecycle, but this ***should not lead to sealed facilities.***

Evolve what it means to be a grid-site instead of replacing them?

- e.g. add interactive options



Access to Data (VO → User)

As a user I need to be able to access collaboration data

- Currently often manual acquisition
- Preference for batch systems w/ shared file system
 - Easy to write code that uses familiar posix semantics with help of lot of existing documentation:

```
rucio download ...  
submit my_analysis \  
--in /sharedfs/home/me/my_analysis/in/*.root \  
--out /sharedfs/home/me/my_analysis/out/*.root
```

*global storage
VO auth / object-y*

*local storage
auth & POSIX*

Access to Data (VO → User)

A priori, it would be nicer, if the user did not have to think about interacting explicitly w/ DDM to get the data

Data Lake

Cache

AF

user job

- Use global namespace to *identify data*, let DDM figure out *how to deliver it*
 - user will expect reliability: if not expect preemptive downloads as a defensive approach
- Then: analysis software cannot purely rely on standard POSIX anymore
 - But e.g. people already comfortable with e.g. `TFile::Open("root:// ... ")`

*Note: mix of
Local and global
storage semantics*

```
submit my_analysis --in user.jsmith.some.dataset  
                    --out /sharedfs/home/me/my_analysis/out/*.root
```

Sharing Data (User → VO)

As a user one must be able to *share data back to the collaboration*.

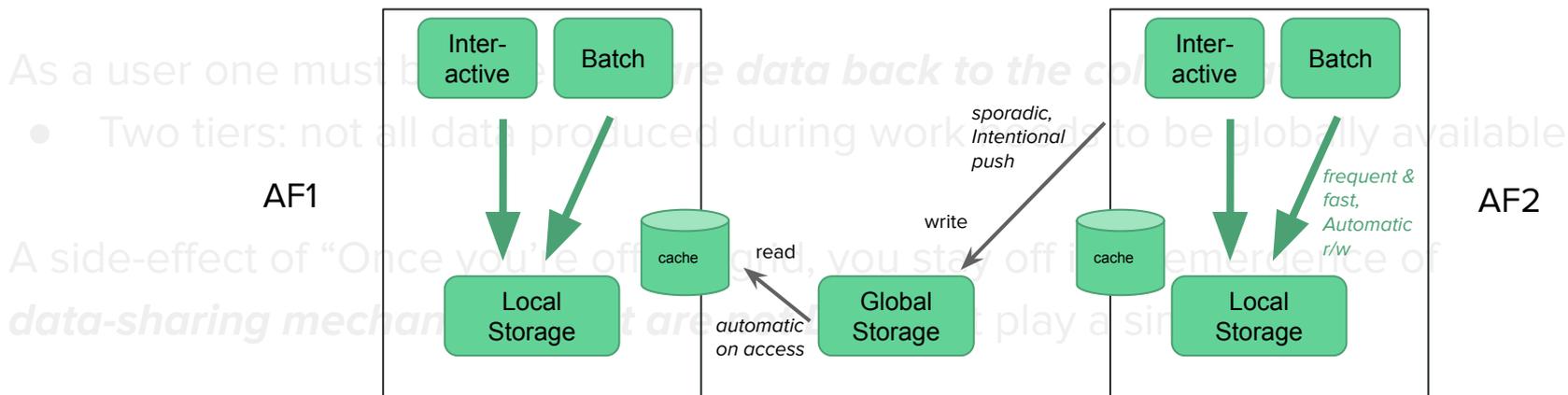
- Two tiers: not all data produced during work needs to be globally available
- Someone *working on AF1* should be able to *share data with someone in AF2*
 - though VO-auth should enable an analysis team to work on a single AF

A side-effect of “Once you’re off the grid, you stay off it” are patterns of *data-sharing mechanisms that are not DDM* but play a similar role:

- Pick a filesystem a majority of analyzers have access & organize analysis data artefacts within a directory structure in that filesystem
- To use those data artefacts analyzers must:
either do their *work at the hosting facility* or *manually copy data from/to it*

Sharing Data (User → VO)

“cp” /sharedfs/home/me/dataproducts*.h5 \
user.lheinric.analysis/dataproducts/



Q: could a **dedicated global group- / analysis- level storage** be possible ?

- don't need instant syncing, unlike jobs, don't need *full* POSIX (à la AFS)
- do need capacity for large scale analysis data artifacts
- Ideally: FUSE-mountable in AF, avoid clobbering w/ r/w from active workloads
 - Worth thinking about who people don't use DDM more, can it be made more user-friendly, is this use-case recognized as in-scope for DDM?

Ability to move facilities

No need to work on N facilities in parallel, **but ability to move facilities** and get a similar experience useful. Solved for grid workloads - but less so for analysis

→ Analysis Facilities become “sticky”

Open Questions revolve around finding common solutions across facilities

- Can one log into the facility ?
- Are services compatible or do I need to change code?
(example: switching analysis code from SLURM to HTCondor painful)
- Uniform user interface (REANA, JupyterHub...) between facilities

Setting up Software

As a user one must be able to set up the **software stack one needs** in the analysis session.

Options discussed:

- Traditional approach **global read-only filesystem (cvmfs) + local files**
 - Has worked quite well so far, but limited access to add s/w and global state exposure
- More recently: user-defined containers
 - Benefits in flexibility, reproducibility, preservation
 - from-scratch container-building not a user-friendly activity
 - well-maintained base images being provided (→ OSG, ATLAS, ...)

Interactive Analysis

Analysis work *lives across a spectrum* of (new R&D \leftrightarrow crank-turning later on) with varying user expectations on *access, availability & turn-around*

Maintaining ability to creatively explore data at HL-LHC is still important and may not fit “single-node”. *Software is being prepared* for this across the board

→ if we want to use these modes, facilities need to support it

```
import dask
import dask_awkward as dak
import hist
import hist.dask as hda
import numpy as np

from coffea import processor
from coffea.nanoevents import NanoEventsFactory

import matplotlib.pyplot as plt

from distributed import Client
client=Client()

# The opendata files are non-standard NanoAOD, so some optional data
processor.NanoAODSchema.warn_missing_crossrefs = False

events = NanoEventsFactory.from_root(
    "file:/Users/lgray/coffea-dev/coffea/Run2012B_SingleMu.root",
    treepath="Events",
    chunks_per_file=500,
    permit_dask=True,
    metadata={"dataset": "SingleMu"}
)
```

dask_histogram + hist

local dask-distributed cluster (

Workers	Coffea 0.7 (EAF)	Coffea 0.7 (GCP)	Coffea 2023	Coffea 2023 (-setup)
1	~1 µs	~1 µs	~1 µs	~1 µs
3	~1 µs	~1 µs	~1 µs	~1 µs
12	~1 µs	~1 µs	~1 µs	~1 µs
24	~1 µs	~1 µs	~1 µs	~1 µs
48	~1 µs	~1 µs	~1 µs	~1 µs

L. Gray Talk at CHEP this week

RDataFrame is going distributed!

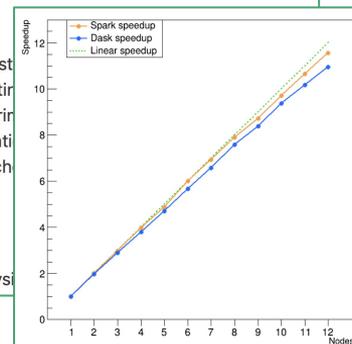
(22 July 2021)

So you love RDataFrame, but would like to use it on a cluster introduced in ROOT a Python package to enable distributed set of remote resources. This feature is available in experimental 6.24 release, allowing users to write and run their applications while steering the computations to, for instance, an Apache

One programming model, many backends

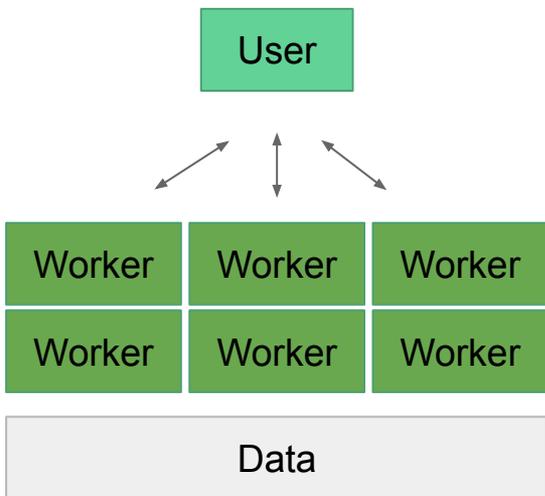
RDataFrame is ROOT's high-level interface for data analysis

<https://root.cern/blog/distributed-rdataframe/>



Interactive Analysis

The typical model is “interactive client + scalable backend”.

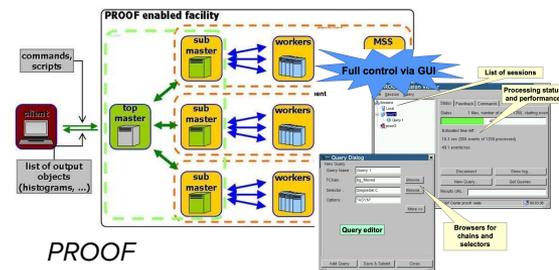


Time-to-session: $O(\text{seconds})$
similar to ssh login

Time-to-first worker: $O(\text{minute})$
short enough to wait & see things are healthy

Time-to-scale up workers: $O(\text{few minutes})$
gradual scale-up is OK

Time-to-finish: $O(30\text{m}-1\text{h})$
anything else isn't really interactive



>20years

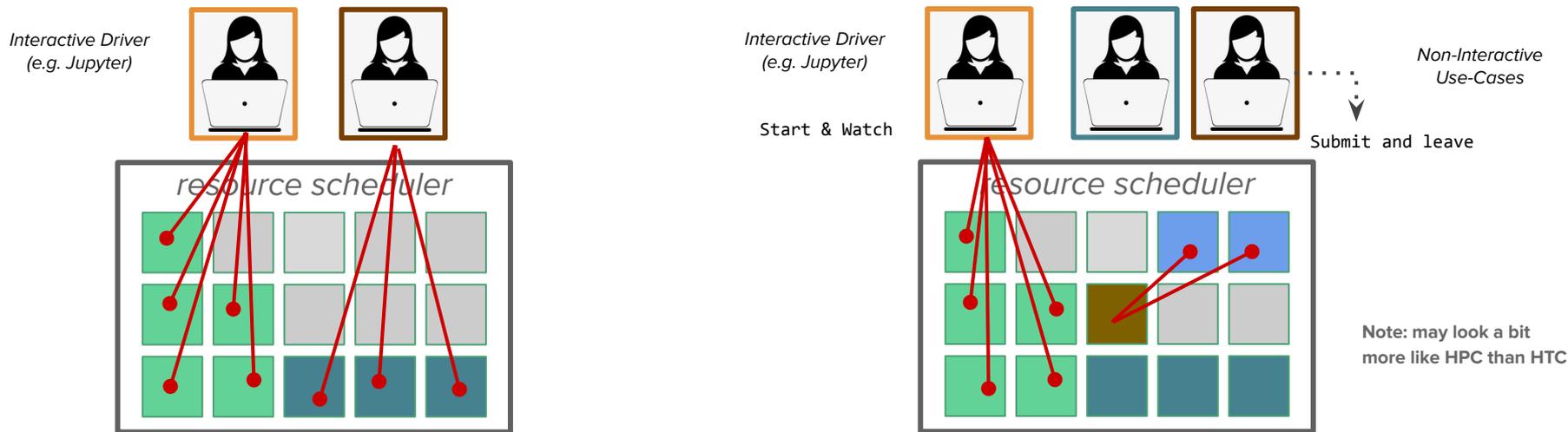


ATLAS/Google Dask Tests (N. Hartmann)

Interactive Analysis - old ideas + new tech. PROOF reborn.

Not everything that starts interactive stays interactive.

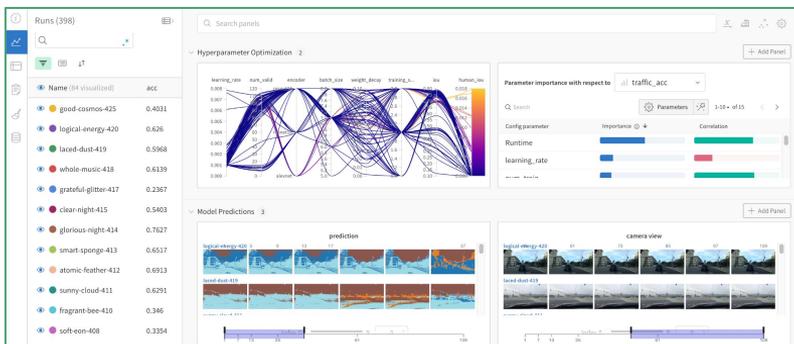
- If scale-out style is common, we need to be able to **schedule such workloads**
- **Not everything is a notebook.** classic mode of terminal + batch must be supported
- A user might not strongly care if it's ssh or some other “terminal as a service”



ML and Heterogeneous Resources

ML is and has been growing in analysis. As a user one expects to be able to exercise the **full ML analysis lifecycle** on a AF

- Data Exploration & Preparation, Interactive R&D and training
- Large-scale non-interactive training and HP optimization
- ML Inference within an analysis pipeline



The screenshot shows a terminal window with a code editor. The code editor contains a document titled 'How to make use of GPU resources with PyTorch.' The code includes a function to check GPU availability and a PyTorch training loop. The terminal output shows the execution of the code, including the command 'python main.py' and the resulting output, which includes the GPU memory usage and the training progress.

ML and Heterogeneous Resources

Software: Interactive & non-interactive analysis infrastructure will support ML workflows as well. Integration of ML-focused tools may be interesting but not crucial

Hardware: main requirement is the availability of GPUs for ***both interactive and non-interactive workloads***

- Asymmetry in utility during interactive phase:
 - 2 User w/ 1 GPUs likely worth more than 1 w/ 2 GPUs
- Time-bound models (use for X min, then get kicked out) work for interactive

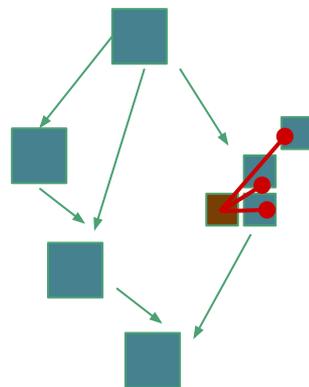
Analysis Preservation & End-to-end Workflows

Much progress in analysis preservation (AP) across experiments. Current mode:

- capture *software in containers*, capture workflow in *workflow languages*

Two discussion points

- What does AP look like if part of workflow is sync-distributed (e.g. Dask)
- Preserved analyses become a tool for analyzers
 - E.g. reinterpretation campaigns
 - Need to be schedulable as workflows
 - workflow as a service



The Way Forward

For the User Requirements section the important questions are:

- Demonstrate rational & demand for listed requirements
- Did we miss anything important ? Your input is important!

Post Workshop:

- Series of meetings to discuss draft → finalize & polish