



CompF4: Analysis Facility Discussion & Priorities

Mark Neubauer, Oksana Shadura, Gordon Watts



Definition of future AF

- What is a definition of nowadays “Analysis Facility” and how we see it for next 5 years?

A working version of an AF definition from the [March HSF AF Forum kick-off meeting](#):

“The infrastructure and services that provide integrated data, software and computational resources to execute one or more elements of an analysis workflow. These resources are shared among members of a virtual organization and supported by that organization.”

How might we think about AFs 5-10 years from now?

Still < 10% of HL-LHC computing resources?

What about a distinction between batch and AF/interactive (ntuply stuff)

- Is batch part of a workflow?
- Neutrion: working on event level, or the ensemble level

What is the definition of the word “workflow” - it contains both interactive and batch elements, timescale of minutes, and days.

Resources: need to share them between DUNE and LHC, for example. Site center support POV: you can’t double the work to support the two experiments. We need a datacenter viewpoint here. Doug B took an action time. (Thanks Doug!)

5-10 years - HW will be different, HPC’s host AF?s But will need HW need to be under Experiment’s control (e.g. through allocations)



Building blocks for AF

- Modern authentication
 - Federated identity!
- Data delivery and data management tools
- Integration of an object storage
- Intensive data caching
- Support of new ecosystems (python) and columnar analysis
- Introducing DevOps techniques for providing
 - Lab vs University environments?
- Reusability/Sharing of existing computing resources
- Shared Usage Controller/Fair Share
 - Fair-share without compromising fast turn-around?
 - More than what a batch system provides?

What we need to add in this list?

Federated centers (multiple centers), integration with Cloud and HPC/Hybrid facilities (components to support that)

Configuration of front-end - so the DUNE user can configure or the LHC user can configure the env they want to use



Analysis facilities and preparation for HL-LHC-era experiments

- Analysis frameworks and its integration in the existing facilities - how to make it successful?
- “Test-driven development” for Analysis Facilities?
 - The 10-minute benchmark?
 - Dev-ops?
 - Other benchmarks (and automation of the benchmarks?)
- Lack of users feedback, “don’t expect tuning and cooperating” - how we can make facilities more user friendly?
 - Monitoring to discover usage patterns?
 - Removing barriers to access
 - Things they need
 - Predictability

Not just HL-LHC, but DUNE too

SWAN - mostly developed old-school way - not iterative development - how well does this work?

US T3 development - lots of survey’s - tests were based on that - use this to feed the (agile) iterative development of the facilities. Workshops to collect people, set up a 50 years workshop where they use the facility all at once to monitor and see how it survives. Aggressively reachout



New features to take in an account while designing Analysis Facilities

- Portability as it is envisioned in L. Bryant talk
 - Building blocks of services
 - Stitched together uniquely for each site
- Interoperability
- Providing Enhanced Security (Doug Benjamin)
- Modern Design Principles (Maria Acosta)
 - Service based architecture
 - Api-driven communication between services
 - Container based infrastructure
 - declarative system to manage the configuration and operations of every element of the platform
 - Where appropriate
- User-defined containers for customizing and sharing python environments (Mark Neubauer)
 - Configuration management, injection of secrets, etc.
- Enabling analysis teams to work together
 - **Shared data**, shared resource usage, shared queries, shared caches, etc.
- Usage Aids
 - Predictability, forecasting how long something will take

How do we manage unmanaged resources (e.g. cleaning up disk, lifetimes of files, etc.). Policy up front?

Sharing between two AF's (code, environment, data). "Enhanced Sharing"

Common portable layer building blocks

Open source building blocks



Discussion Points

Interoperability

- Labs vs Universities
 - Putting services together that are individually managed and bespoke
 - Putting together a chart that spins up everything (ensemble of services designed as a single installation unit)
 - Can we create a common experience in an AF for all users with these two approaches? Should we?
- What do we use as the substrate - how do we automate our infrastructure?
 - Universities - K8S
 - Labs, other places - OpenShift
- Different cloud providers?
 - University, Lab, Amazon, Azure, Google?
- Experiment environment vs Modern Python Environment - Compatibility?
 - Experiment environment (C++ development)
 - Modern analysis environment (newest ROOT, python, DASK, etc.)
 - Containers with same view of shared disk resources (user/scratch)?

How do we make sure we don't have islands of development?