Building a Columnar Analysis Demonstrator for ATLAS PHYSLITE Open Data using the Python Ecosystem

KyungEon Choi, Matthew Feickert, Nikolai Hartmann, Lukas Heinrich, Alexander Held, Evangelos Kourlitis, Nils Krumnack, Giordon Stark, Matthias Vigl, Gordon Watts on behalf of the **ATLAS Computing Activity**
(University of Wisconsin-Madison)
matthew.feickert@cern.ch

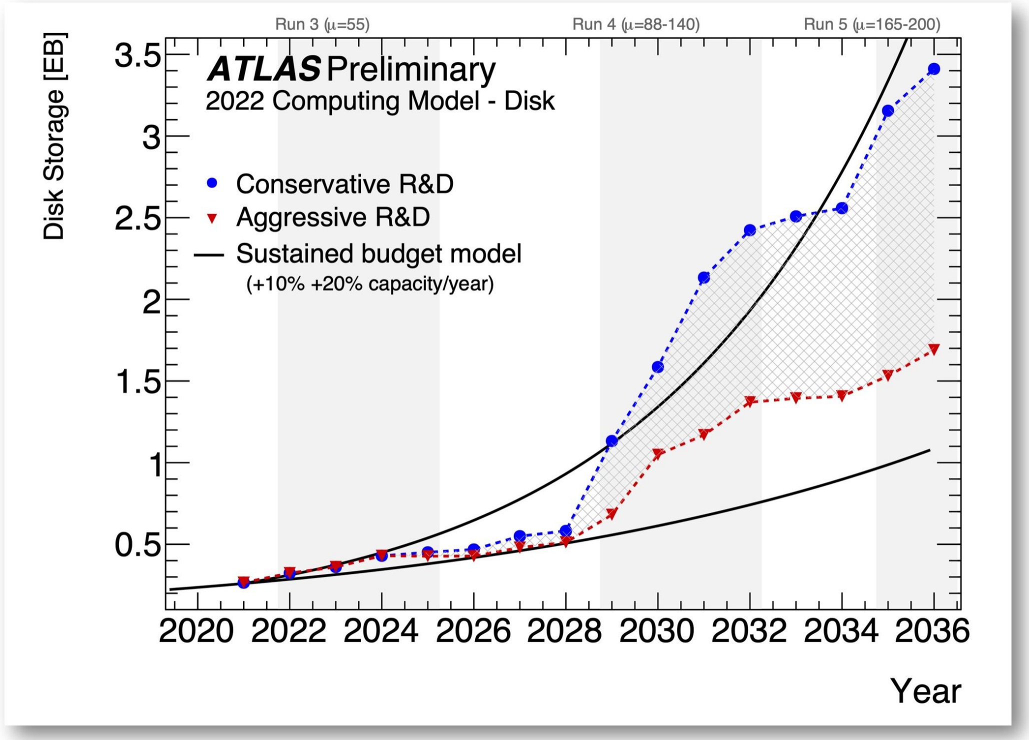
International Conference on Computing in High Energy and Nuclear Physics (CHEP) 2024
October 21st, 2024

Data Science Institute
UNIVERSITY OF WISCONSIN–MADISON

iris hep

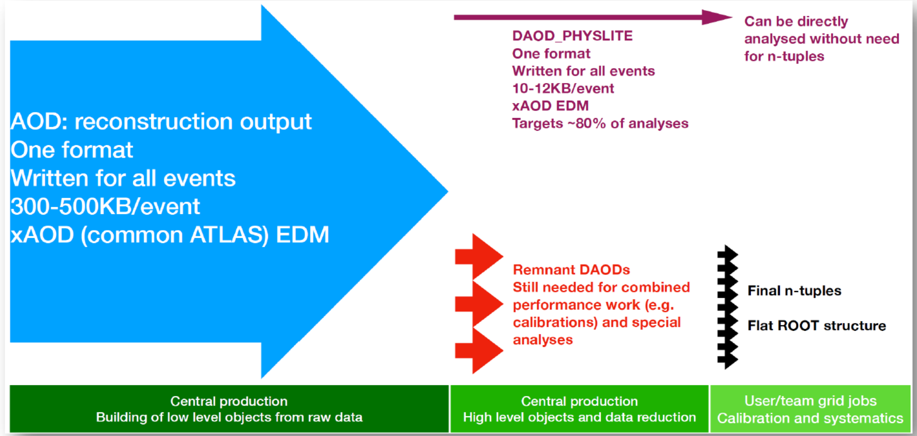
ATLAS EXPERIMENT

Challenges for Future Analysis



(ATLAS Software and Computing HL-LHC Roadmap, 2022)

- Won't be able to store everything on disk
- Move towards "trade disk for CPU" model


AOD: reconstruction output
One format
Written for all events
300-500KB/event
xAOD (common ATLAS) EDM
DAOD_PHYSLITE
One format
Written for all events
10-12KB/event
xAOD EDM
Targets ~80% of analyses
Can be directly analysed without need for n-tuples
Remnant DAODs
Still needed for combined performance work (e.g. calibrations) and special analyses
Final n-tuples
Flat ROOT structure
Central production
Building of low level objects from raw data
Central production
High level objects and data reduction
User/team grid jobs
Calibration and systematics

(Jana Schaarschmidt, CHEP 2023)

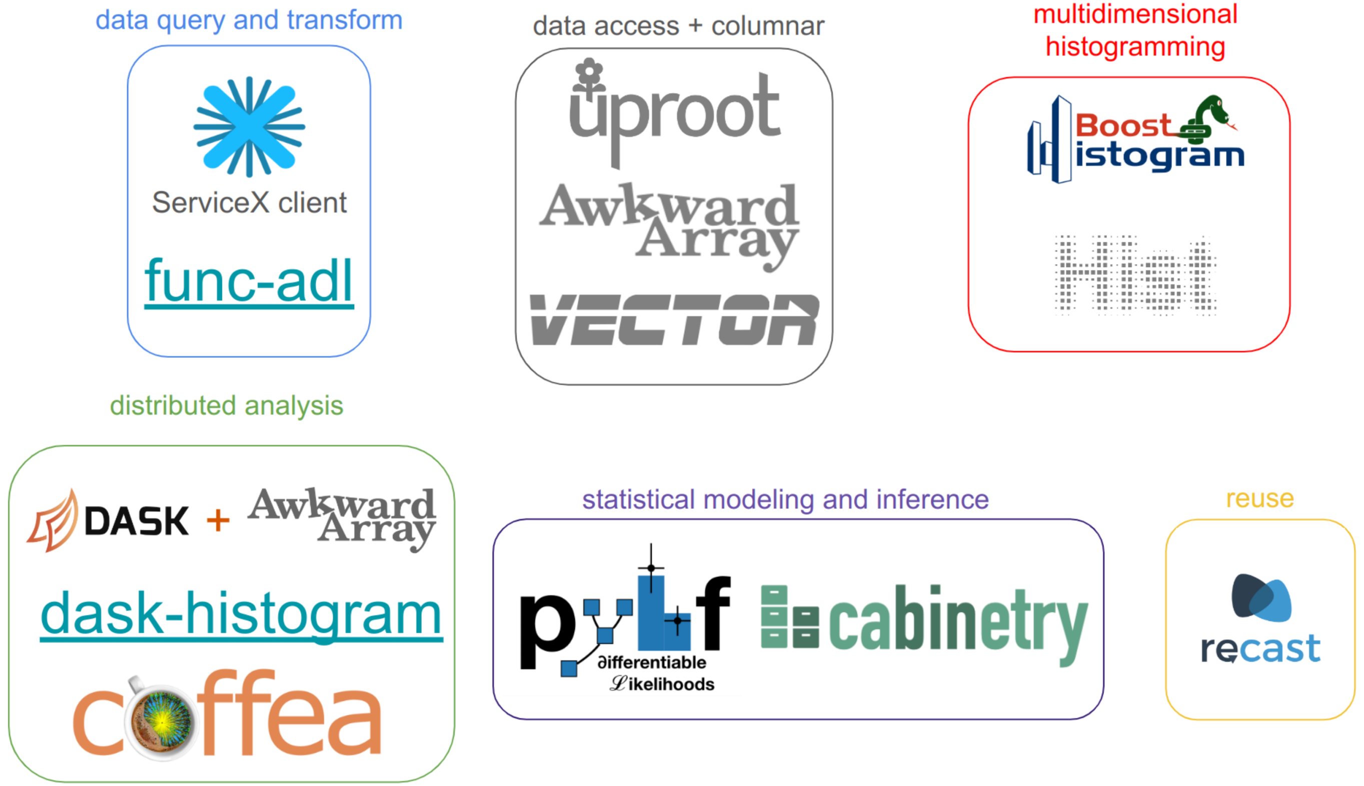
PHYSLITE

- Common file format for **Run 4 Analysis Model**
- Monolithic: Intended to serve ~80% of physics analysis in Run 4
- Contains already-calibrated objects for fast analysis
- Will be able to use directly without need for ntuples

Pythonic Ecosystem for ATLAS Analysis

Providing the elements of a **columnar analysis pipeline**

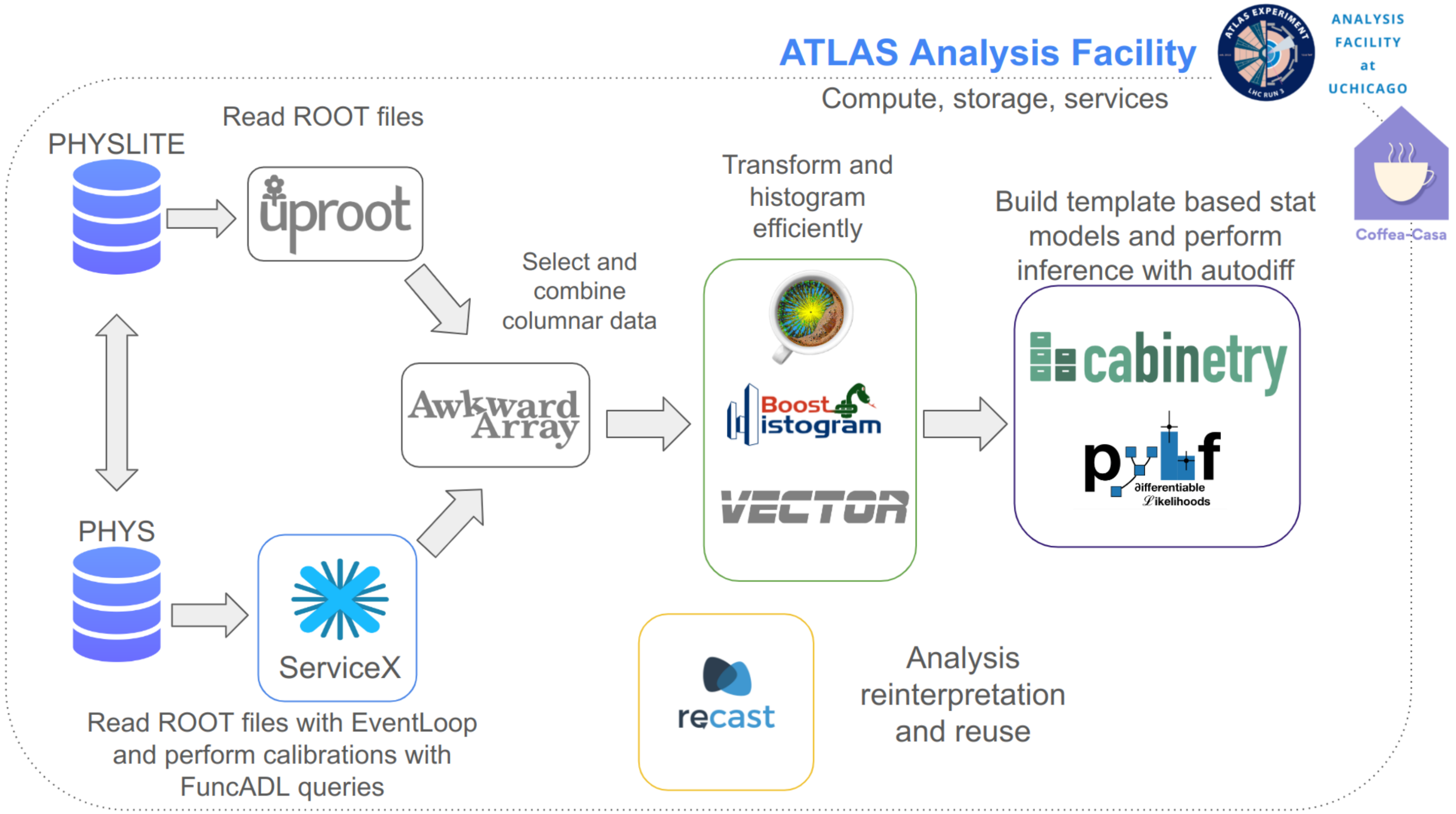
- Data query and access
- Reading data files (ROOT and others) and columnar access
- Data transformation and histogramming
- Distributed analysis frameworks
- Statistical inference
- Analysis reinterpretation



Composing structure of an ATLAS AGC

End user analysis ideally uses **smaller and calibrated PHYSLITE**

Can still use PHYS (same data format) through will need to perform **additional steps** (calibration) with funcADL

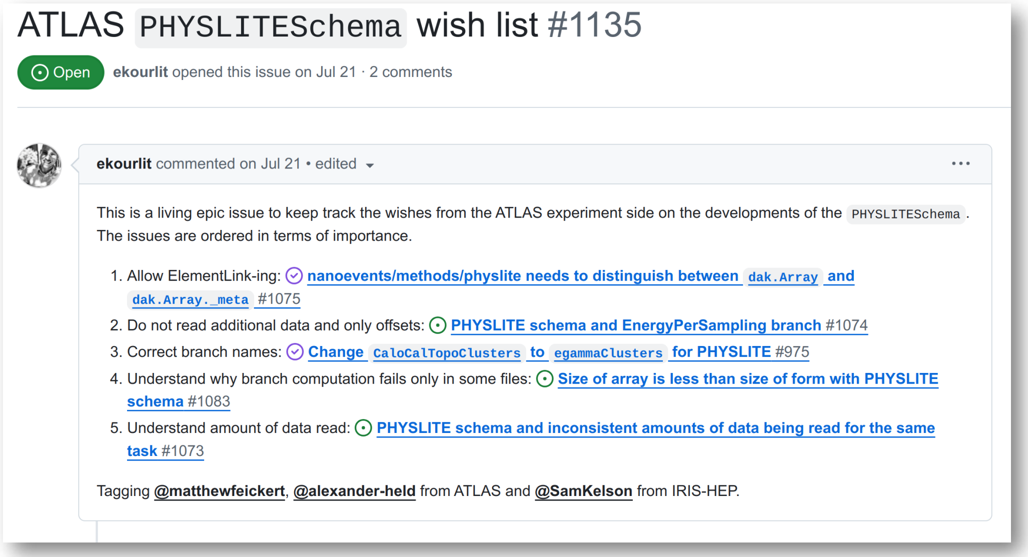


ATLAS Analysis Facility
Compute, storage, services
ANALYSIS FACILITY at UCHICAGO
Coffea-Casa
PHYSLITE
Read ROOT files
uproot
Select and combine columnar data
Awkward Array
Transform and histogram efficiently
Boost histogram
VECTOR
Build template based stat models and perform inference with autodiff
cabinetry
pyhf
differentiable likelihoods
PHYS
ServiceX
Read ROOT files with EventLoop and perform calibrations with FuncADL queries
recast
Analysis reinterpretation and reuse

Components of an ATLAS Analysis Grand Challenge (AGC) demonstrator pipeline (c.f. The 200Gbps Challenge (Alexander Held, Monday plenary))

Challenges: Reading all PHYSLITE files

- Raw PHYSLITE is not easily loadable by columnar analysis tools outside of ROOT
  - Challenges for correctly handling `ElementLinks` and custom objects (e.g. triggers)
- Awkward Array supports `behaviors`, which allow for efficiently reinterpreting data on the fly
- ATLAS members have contributed to open ecosystem development to support PHYSLITE in both Uproot and Coffea
- Continuing to support fixes to both the PyHEP ecosystem tools as well as reporting issues to PHYSLITE
  - Work by ATLAS IRIS-HEP Fellow Sam Kelson



ATLAS `PHYSLITESchema` wish list #1135

Open **ekourlit** opened this issue on Jul 21 · 2 comments

ekourlit commented on Jul 21 • edited

This is a living epic issue to keep track the wishes from the ATLAS experiment side on the developments of the `PHYSLITESchema`. The issues are ordered in terms of importance.

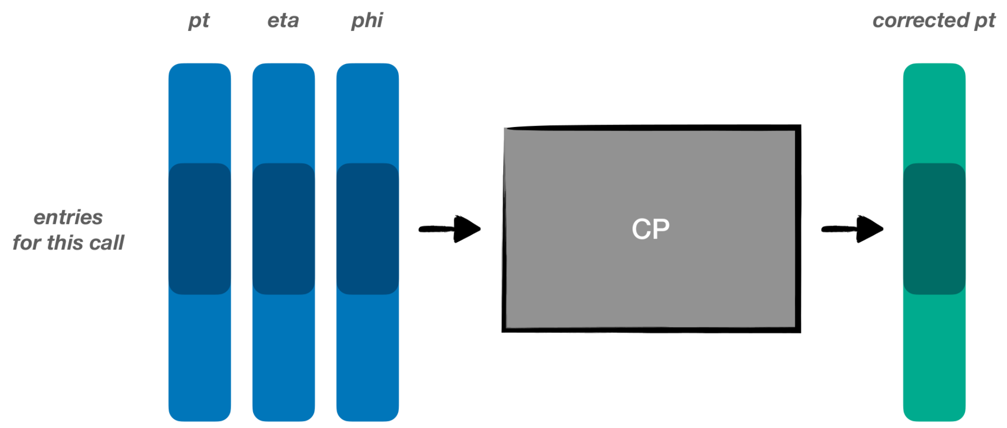
1. Allow ElementLink-ing: nanoevents/methods/physlite needs to distinguish between `dak.Array` and `dak.Array._meta` #1075
2. Do not read additional data and only offsets: PHYSLITE schema and EnergyPerSampling branch #1074
3. Correct branch names: Change `CaloCalTopoClusters` to `egammaClusters` for PHYSLITE #975
4. Understand why branch computation fails only in some files: Size of array is less than size of form with PHYSLITE schema #1083
5. Understand amount of data read: PHYSLITE schema and inconsistent amounts of data being read for the same task #1073

Tagging **@matthewfeickert**, **@alexander-held** from ATLAS and **@SamKelson** from IRIS-HEP.

- More on ATLAS Open Data at CHEP 2024:
  - The First Release of ATLAS Open Data for Research (Zach Marshall, Monday plenary)
  - Open Data at ATLAS: Bringing TeV collisions to the World (Giovanni Guerrieri, Monday Track 8)

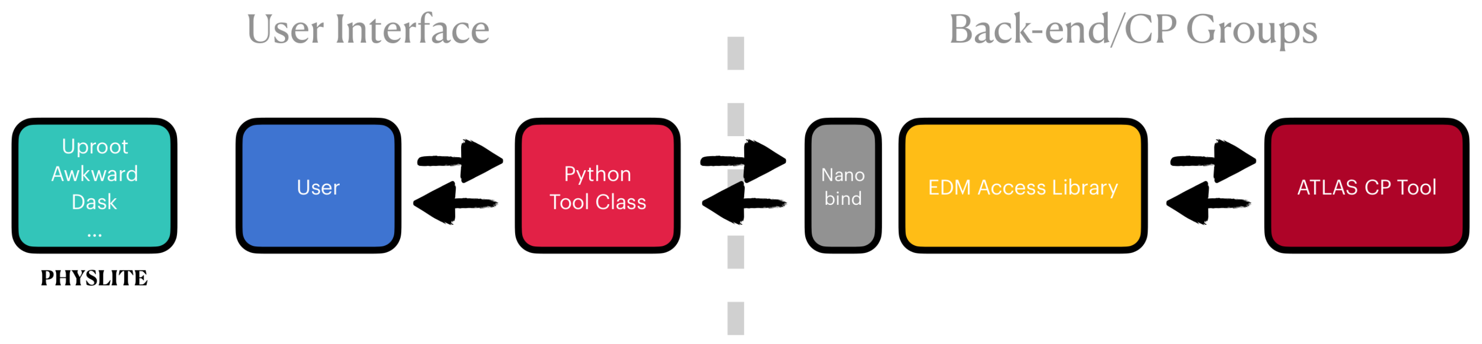
Challenges: Systematics

- As columnar analysis **processes events in batches** also need CP tools and algorithms to process in batches
- Current CP tools operate on xAOD event data model (EDM) for calculation and write systematics to disk for future access (I/O heavy)
- Challenge: Can we adapt this model to work in on-the-fly computation columnar paradigm?
    - Help with "trade disk for CPU" model
- Refactoring tools to a columnar backend in ATLAS show **improvements in performance and flexibility**

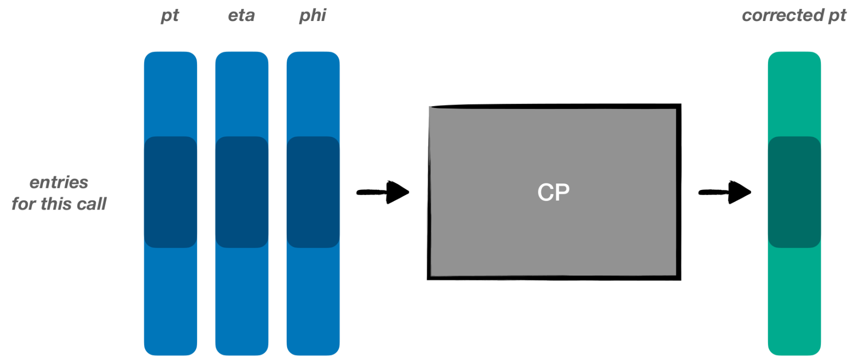


Columnar combined performance (CP) tools operate on existing columns in batches to generate new columns (Matthias Vigl, ACAT 2024)

Challenges and Opportunities: Systematics



- Refactoring to columnar CP tools has allowed for **more Pythonic** array interfaces to be developed
- Using next generation of C++/Python binding libraries allows
  - Zero-copy operations to/from n-dimensional array libraries in Python that supports GPUs
  - Full design control of high-level user API (unified UX)



```
electrons.pt = energyCorrectionTool(electrons,sys="Res_up").newPt
```

Columnar combined performance (CP) tools operate on existing columns in batches to generate new columns (Matthias Vigl, ACAT 2024)

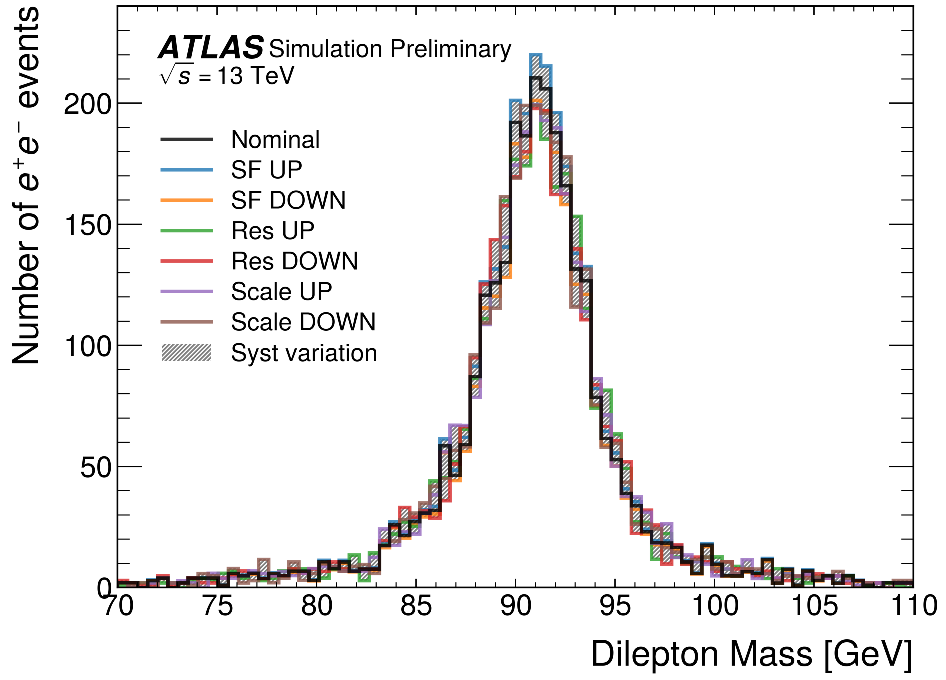
Columnar CP tools: $Z \rightarrow e^+e^-$ Demo

Demo of prototype (v1) zero-copy Python bindings to columnar Egamma CP tool to compute systematics on the fly for m_{ee}.

1. Use Uproot to load PHYSLITE Monte Carlo into Awkward arrays
2. Apply selections with Uproot and Coffea
3. Initialize tools

```
from atlascp import EgammaTools
```

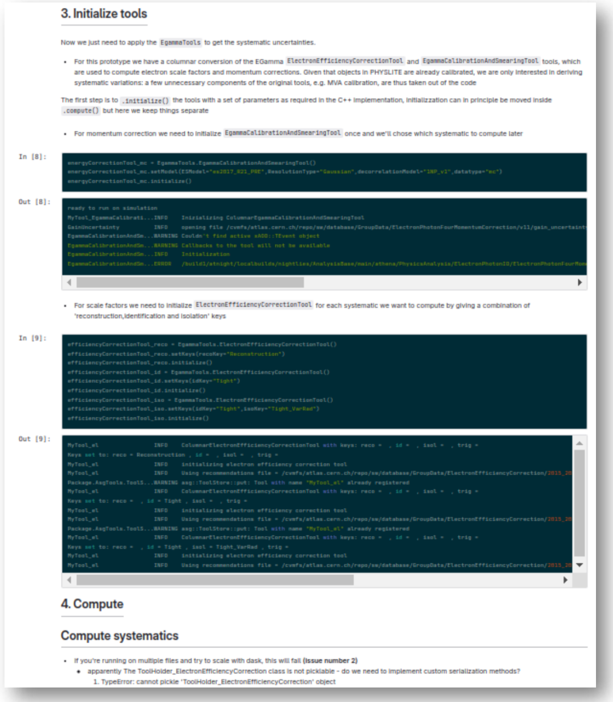
4. Compute systematics on the fly efficiently scaled with dask-awkward on UChicago ATLAS Analysis Facility



Selected m_{ee} under on-the-fly computed systematic variations of electron reconstruction efficiency and corrections
(Matthias Vigl, ACAT 2024)

Iteratively moving columnar tools forward

- v1 prototype established foundations of what was possible with new tooling
  - Pythonic interfaces to CP tools could be written without heroic levels of work
  - Prototype tools were promising, but more work needed to achieve necessary performance
  - No "zero action" option — needed to create standalone prototype to determine if work was reasonable
- v2 prototype takes a step forward in scope
  - Moves developments into ATLAS Athena and **migrate ATLAS CP tools to columnar backend** without breaking existing workflows
    - Adds thread-safety
  - Adds infrastructure support for development of columnar analysis tools
  - Allows for full scale integration and performance tests



3. Initialize tools

Now we just need to apply the `EgammaTools` to get the systematic uncertainties.

- For this prototype we have a columnar conversion of the EGamma `ElectronEfficiencyCorrectionTool` and `EgammaCalibrationAndSmearingTool` tools, which are used to compute electron scale factors and momentum corrections. Given that objects in PHYSLITE are already calibrated, we are only interested in deriving systematic variations: a few unnecessary components of the original tools, e.g. MVA calibration, are thus taken out of the code

The first step is to `.initialize()` the tools with a set of parameters as required in the C++ implementation, initializzation can in principle be moved inside `.compute()` but here we keep things separate

- For momentum correction we need to initialize `EgammaCalibrationAndSmearingTool` once and we'll chose which systematic to compute later

In [8]:
```
energyCorrectionTool_mc = EgammaTools.EgammaCalibrationAndSmearingTool()
energyCorrectionTool_mc.setModel(ESModel="es2017_R21_PRE",ResolutionType="Gaussian",decorrelationModel="1NP_v1",datatype="mc")
energyCorrectionTool_mc.initialize()
```

Out [8]:
```
ready to run on simulation
MyTool_EgammaCalibrati...INFO    Inizializing ColumnarEgammaCalibrationAndSmearingTool
GainUncertainty          INFO    opening file /cvmfs/atlas.cern.ch/repo/sw/database/GroupData/ElectronPhotonFourMomentumCorrection/v11/gain_uncertaint
EgammaCalibrationAndSm...WARNING Couldn't find active xAOD::TEvent object
EgammaCalibrationAndSm...WARNING Callbacks to the tool will not be available
EgammaCalibrationAndSm...INFO    Initialization
EgammaCalibrationAndSm...ERROR   /build1/atnight/localbuilds/nightlies/AnalysisBase/main/athena/PhysicsAnalysis/ElectronPhotonID/ElectronPhotonFourMom
```

- For scale factors we need to initialize `ElectronEfficiencyCorrectionTool` for each systematic we want to compute by giving a combination of 'reconstruction,identification and isolation' keys

In [9]:
```
efficiencyCorrectionTool_reco = EgammaTools.ElectronEfficiencyCorrectionTool()
efficiencyCorrectionTool_reco.setKeys(recoKey="Reconstruction")
efficiencyCorrectionTool_reco.initialize()
efficiencyCorrectionTool_id = EgammaTools.ElectronEfficiencyCorrectionTool()
efficiencyCorrectionTool_id.setKeys(idKey="Tight")
efficiencyCorrectionTool_id.initialize()
efficiencyCorrectionTool_iso = EgammaTools.ElectronEfficiencyCorrectionTool()
efficiencyCorrectionTool_iso.setKeys(idKey="Tight",isoKey="Tight_VarRad")
efficiencyCorrectionTool_iso.initialize()
```

Out [9]:
```
MyTool_el                INFO    ColumnarElectronEfficiencyCorrectionTool with keys: reco =  , id =  , isol =  , trig =
Keys set to: reco = Reconstruction , id =  , isol =  , trig =
MyTool_el                INFO    initializing electron efficiency correction tool
MyTool_el                INFO    Using recommendations file = /cvmfs/atlas.cern.ch/repo/sw/database/GroupData/ElectronEfficiencyCorrection/2015_20
Package.AsgTools.ToolS...WARNING asg::ToolStore::put: Tool with name "MyTool_el" already registered
MyTool_el                INFO    ColumnarElectronEfficiencyCorrectionTool with keys: reco =  , id =  , isol =  , trig =
Keys set to: reco =  , id = Tight , isol =  , trig =
MyTool_el                INFO    initializing electron efficiency correction tool
MyTool_el                INFO    Using recommendations file = /cvmfs/atlas.cern.ch/repo/sw/database/GroupData/ElectronEfficiencyCorrection/2015_20
Package.AsgTools.ToolS...WARNING asg::ToolStore::put: Tool with name "MyTool_el" already registered
MyTool_el                INFO    ColumnarElectronEfficiencyCorrectionTool with keys: reco =  , id =  , isol =  , trig =
Keys set to: reco =  , id = Tight , isol = Tight_VarRad , trig =
MyTool_el                INFO    initializing electron efficiency correction tool
MyTool_el                INFO    Using recommendations file = /cvmfs/atlas.cern.ch/repo/sw/database/GroupData/ElectronEfficiencyCorrection/2015_20
```

4. Compute

Compute systematics

- If you're running on multiple files and try to scale with dask, this will fail **(issue number 2)**
  - apparently The ToolHolder_ElectronEfficiencyCorrection class is not picklable - do we need to implement custom serialization methods?
    1. TypeError: cannot pickle 'ToolHolder_ElectronEfficiencyCorrection' object

ATLAS EXPERIMENT Columnar Athena

Columnar CP tool backend performance tests

- During (ongoing) refactor added preliminary integrated benchmark to measure **time spent in tool per event** (not I/O) and compare to xAOD model
- While direct comparison not possible, tests are as close as possible
  - Only involves `C++` CP tool code (no Python involved)
  - Uses same version of CP tool
  - xAOD includes event store access (per-event overhead, paid per-batch in columnar)
- Show **substantial speedups** for migrated tools: **columnar is 2-4x faster** than xAOD interface (EDM access dependent)
  - Time for I/O and connecting columns not included in the performance comparisons (not optimized in the tests, so removed from benchmark)

Challenges: Tooling design decisions

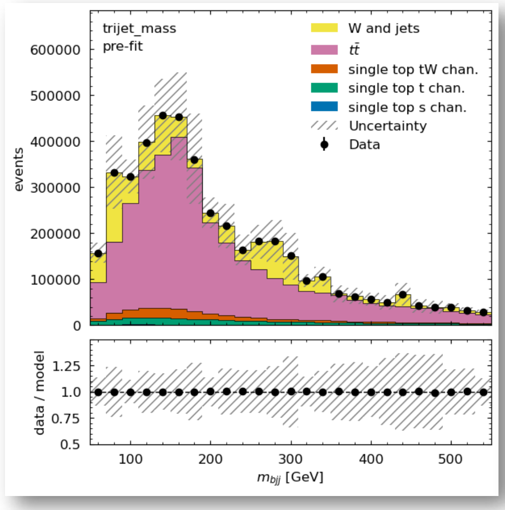
- ATLAS CP tools were created 10-15 years ago to **run in an analysis framework**
  - Battle tested, extremely well understood, excellent physics performance, strong desire to be maintained
  - Rewrite cost is currently too high across collaboration to move to `correctionlib` paradigm
  - Columnar **cracks open "black box"** implementations of tools for the new analysis model
  - Legacy code decisions highlight columnar prototype design decisions and opportunities during tool migration
- Raises the question: "What would it take to get to **`python -m pip install atlascp`**?"
  - Ambitious idea not as far fetched as you might think: `pip install ROOT` (Vincenzo Padulano, Monday Track 6)
- Columnar prototype explores these possibilities
  - **Adopting columnar backend** makes columnar paradigm possible
  - **Ongoing `nanobind` integration** bridges `C++`/Python with performance
  - **Pythonic API design** for high level analysis thinking
- Steps beyond: Modularization to level that allows packaging with `scikit-build-core`
  - Allows for "just another" tool in the PyHEP ecosystem

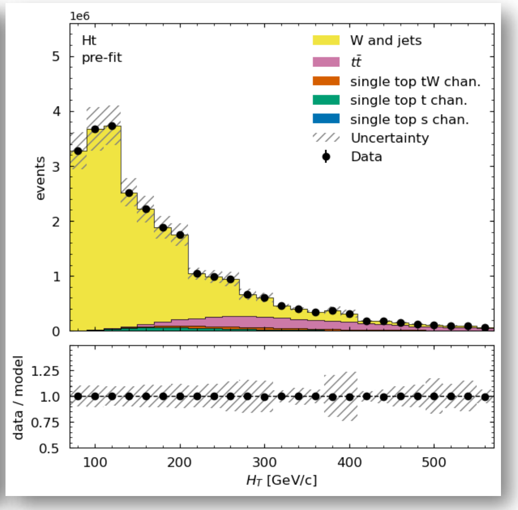
ATLAS Open Data AGC Implementations

- Tooling ecosystem is proving **approachable and performant** for Pythonic columnar analysis of PHYSLITE
- Enabling mentored university students to implement versions of the AGC by themselves in a Jupyter notebook
- ATLAS IRIS-HEP Fellow Denys Klekots's AGC project using **ATLAS open data** (implementation on GitHub)
- Simplified version of IRIS-HEP AGC top reconstruction challenge using 2015+2016 Run 2 Monte Carlo from the 2024 **ATLAS open data** release

Event selection

- 1 charged lepton
- ≥ 4 hadronic jets
- Lepton kinematics: $p_T \geq 30\ \mathrm{GeV}, |\eta| < 2.1$
- Jet kinematics: $p_T \geq 25\ \mathrm{GeV}, |\eta| < 2.4$





ATLAS open data

Summary of ATLAS Columnar AGC Efforts

- Columnar analysis tool efforts inside of ATLAS have been promising with CP tools showing performance increases and bespoke UI
- Development of a columnar ATLAS AGC demonstrator with full systematics is ongoing supported by advancements in v2 prototype
- Technical advancements are being incorporated into ATLAS wide tooling
- Contributions upstream to PyHEP community tools
- ATLAS Open Data proving to be useful for research and community communication
- Advancements in tooling are enabling researchers across career stages

Acknowledgements

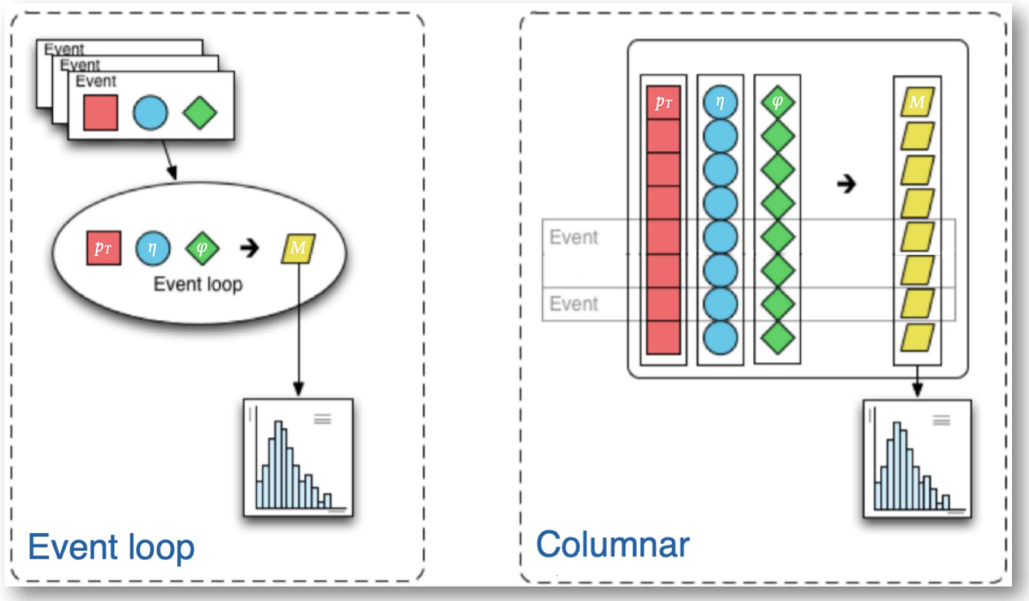
This work was supported in part by the United States National Science Foundation under Cooperative Agreements OAC-1836650 and PHY-2323298 (Institute for Research and Innovation in Software for High Energy Physics (IRIS-HEP)).

Backup

Columnar Analysis

"columnar analysis" == "array programming for data analysis"

- Higher level APIs for physicists and improved user experience
  - People using columnar analysis on ntuples already seem to be loving it
  - Enable the same UX but without ntupling (save disk)
- Potential for higher performance
  - Enable on-the-fly combined performance (CP) tool corrections on PHYSLITE
- Broader scientific data analysis ecosystem integration
  - Extend and scale ATLAS tools with large and performant ecosystem



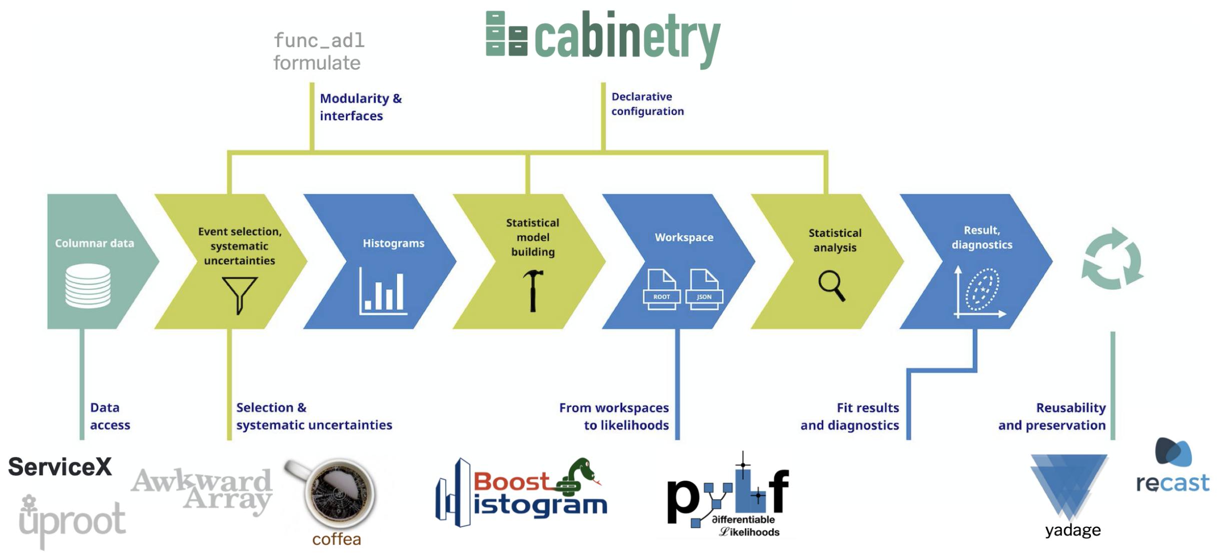
Different expressions/representations for same analysis result goals

(Nick Smith, 2019 Joint HSF/OSG/WLCG Workshop)

An Analysis Grand Challenge

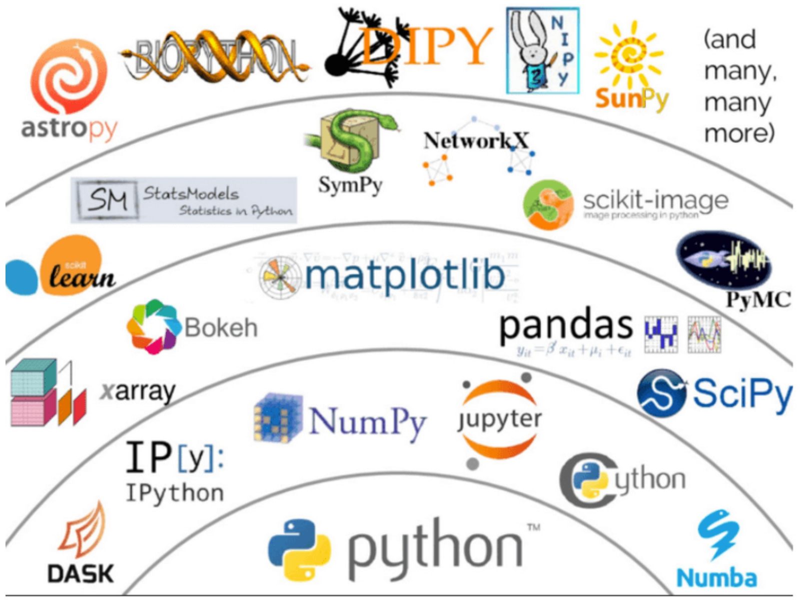
HL-LHC era data scale requires rethinking interacting with data during analysis

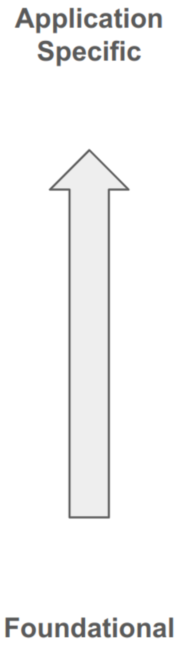
- **Analysis Grand Challenge** (AGC) community exercise organized by IRIS-HEP includes the stages of a projected typical HL-LHC analysis
- Demonstrator of development of the required cyberinfrastructure
  - The 200Gbps Challenge: Imagining HL-LHC analysis facilities (Alexander Held, Monday plenary)
- Opportunity for ATLAS to demonstrate columnar analysis views and areas for improvement

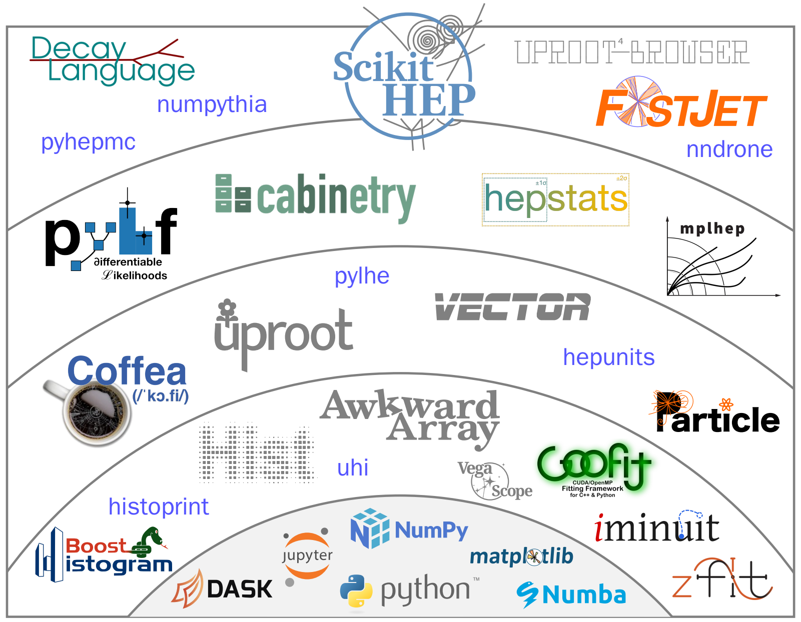

func_adl
formulate
cabinetry
Modularity & interfaces
Declarative configuration
Columnar data
Event selection, systematic uncertainties
Histograms
Statistical model building
Workspace
Statistical analysis
Result, diagnostics
Data access
Selection & systematic uncertainties
From workspaces to likelihoods
Fit results and diagnostics
Reusability and preservation
ServiceX
uproot
Awkward Array
coffea
Boost-histogram
pyhf
differentiable likelihoods
yadage
recast

High level view of operations in an HL-LHC analysis

Pythonic Analysis Ecosystem for HEP





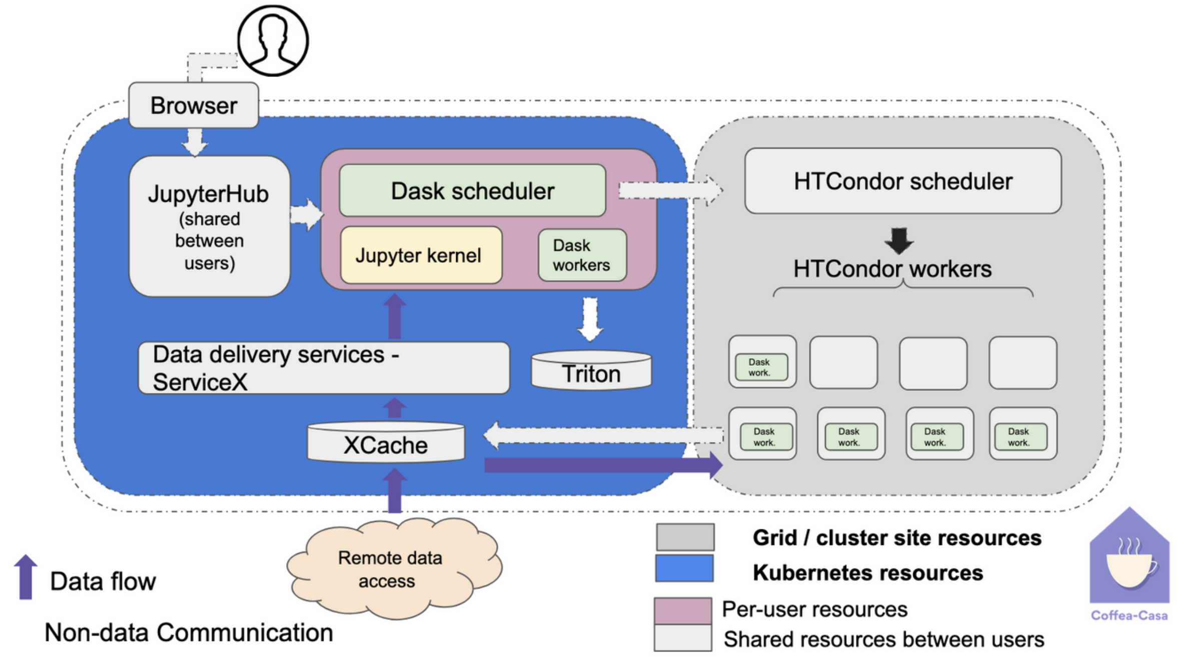


Broader "Scientific Python" ecosystem is designed to be interoperable and support multiple domain levels

Interoperable domain hierarchy design continued in "PyHEP" ecosystem

Prototyping on US ATLAS Analysis Facilities

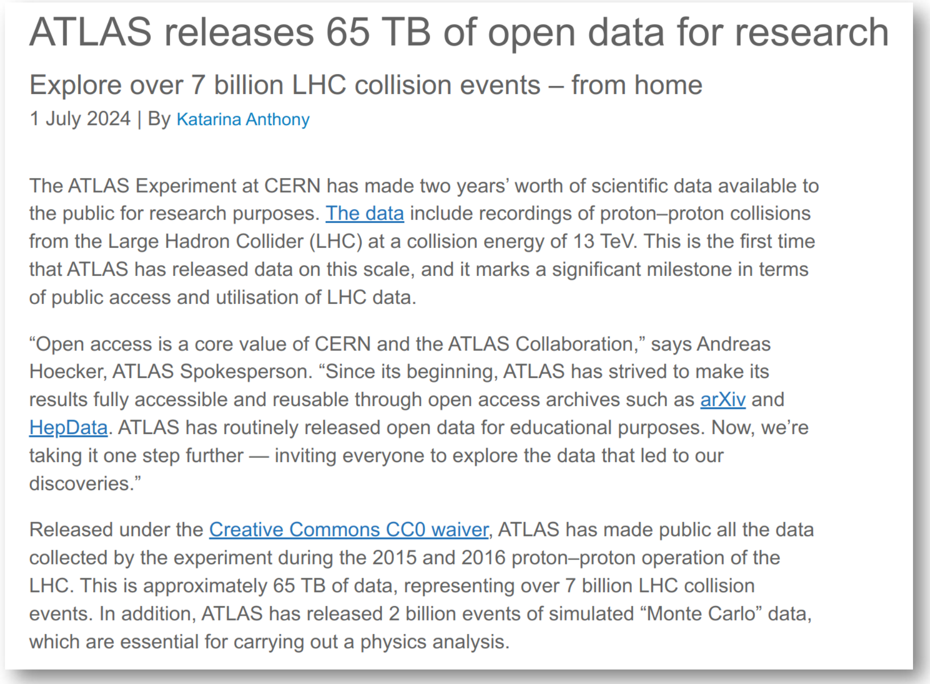
- University of Chicago Analysis Facility **provides testing bed** for analysis platform
- Provides support for:
  - **JupyterLab** as a common interface
  - Highly efficient data delivery with **XCache**
  - Conversion to columnar formats with **ServiceX**
- Excellent integration exercise between analysis and operations



Scalable platform for interactive (or noninteractive) analysis

ATLAS Open Data

- **First** release of ATLAS Run 2 2015 and 2016 open data in July 2024
- Using ATLAS open data for AGC
    - Open access data allows for use in testing community projects and problems
    - Released as PHYSLITE (HL-LHC data format)
    - Allows for new students to be able to learn analysis and make contributions quickly
- More on ATLAS Open Data at CHEP 2024:
    - The First Release of ATLAS Open Data for Research (Zach Marshall, Monday plenary)
    - Open Data at ATLAS: Bringing TeV collisions to the World (Giovanni Guerrieri, Monday Track 8)



ATLAS releases 65 TB of open data for research

Explore over 7 billion LHC collision events – from home

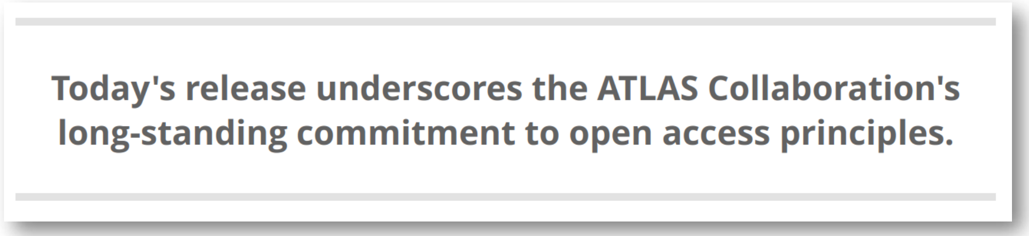
1 July 2024 | By Katarina Anthony

The ATLAS Experiment at CERN has made two years' worth of scientific data available to the public for research purposes. The data include recordings of proton–proton collisions from the Large Hadron Collider (LHC) at a collision energy of 13 TeV. This is the first time that ATLAS has released data on this scale, and it marks a significant milestone in terms of public access and utilisation of LHC data.

"Open access is a core value of CERN and the ATLAS Collaboration," says Andreas Hoecker, ATLAS Spokesperson. "Since its beginning, ATLAS has strived to make its results fully accessible and reusable through open access archives such as arXiv and HepData. ATLAS has routinely released open data for educational purposes. Now, we're taking it one step further — inviting everyone to explore the data that led to our discoveries."

Released under the Creative Commons CC0 waiver, ATLAS has made public all the data collected by the experiment during the 2015 and 2016 proton–proton operation of the LHC. This is approximately 65 TB of data, representing over 7 billion LHC collision events. In addition, ATLAS has released 2 billion events of simulated "Monte Carlo" data, which are essential for carrying out a physics analysis.

(ATLAS News, 2024-07-01)



Today's release underscores the ATLAS Collaboration's long-standing commitment to open access principles.

Reading PHYSLITE with Columnar Backends

- For reading `ElementLink` and other unreadable members are pursuing multiple strategies
- Have Awkward behaviors in Python, but we also try to turn everything into "plain old data" (POD) branches, and RNTuple will help with that
- If only target infrastructure was Uproot we could stick with Awkward behaviors, but RDF (without dictionaries), and Julia would also have to support such custom reading, and that's not a scalable approach

References

- ATLAS Software and Computing HL-LHC Roadmap, ATLAS Collaboration, 2022
- Documentation on PHYSLITE Variables for ATLAS Open Data, ATLAS Collaboration, Accessed 2024
- Using Legacy ATLAS C++ Calibration Tools in Modern Columnar Analysis Environments, Matthias Vigl, ACAT 2024
- How the Scientific Python ecosystem helps answering fundamental questions of the Universe, Vangelis Kourlitis, Matthew Feickert, and Gordon Watts, SciPy 2024
- ATLAS PHYSLITE Content Documentation, ATLAS Collaboration, Accessed 2024 [ATLAS Internal]
- The Columnar Analysis Grand Challenge Demonstrator, Gordon Watts, ATLAS S&C Plenary Afternoon: Demonstrators, 2023-10-04 [ATLAS Internal]
- ATLAS AGC Demonstrator, Gordon Watts, ATLAS AMG+ADC Joint Session, 2023-03-30 [ATLAS Internal]
- Tour of the CP Columnar Prototype and CP Algorithm Conversion, Nils Krumnack, 2024-10-07 [ATLAS Internal]