Brookhaven National Laboratory

ATLAS, HL-LHC and IRIS-HEP plans for analysis

Johannes Elmsheuser

25 June 2021, NPPS tech meeting #8 - Analysis tools & services

Material

- HL-LHC Analysis Mini-Workshop, 4 May, indico agenda
- IRIS-HEP Analysis Systems: webpage link
- ”Advances in Analysis tools/ecosystem”, O. Shadura at LHCP conference, talk link (Copied several slides)
- ”Analysis Tools and New workflows”, T. Maeno at ATLAS S&C week, talk link (Copied several slides)
- ATLAS HL-LHC Computing Conceptual Design Report, CERN-LHCC-2020-015, esp. Section 8
- ATLAS CPU and Disk resource projection plots, see link
- Evolution of the ATLAS analysis model for Run-3 and prospects for HL-LHC, https://doi.org/10.1051/epjconf/202024506014
- Series of meetings about ”Analysis software in the wider HEP/nuclear community” in HSF DAWG (indico category)

LHC TIMELINE

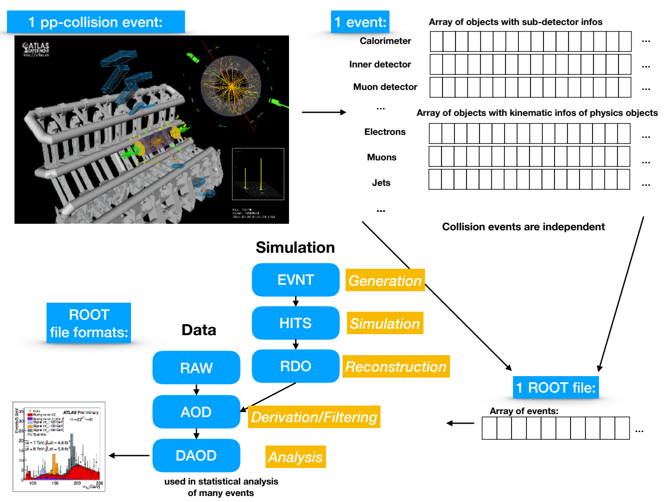
Outline

User analysis in the ATLAS experiment

IRIS-HEP

ROOT

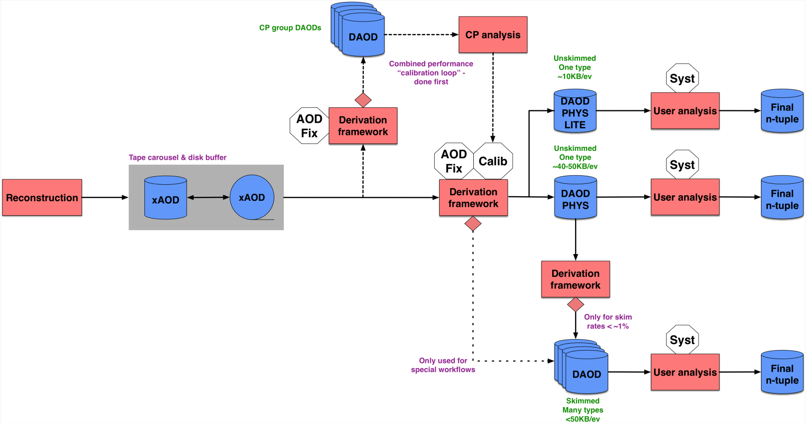
Introduction: Simplified data analysis workflow for ATLAS



In essence: several steps of data processing and then **data reduction**

First parts on Grid/Cloud/HPC - last step usually on local resources

Run3 analysis production workflows and formats



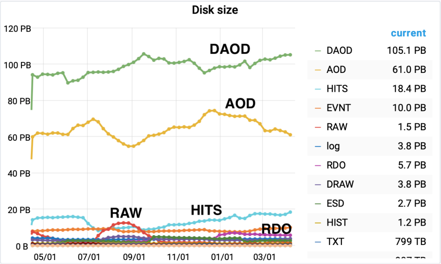
DAOD_PHYS:
50 kB/event, combined single DAOD format (for MC, but also DATA), AOD event data model (EDM)

DAOD_PHYSLITE:
10 kB/event, very condensed and calibrated objects, very important for HL-LHC, AOD or ntuple EDM, ideal for DOMA/XCache

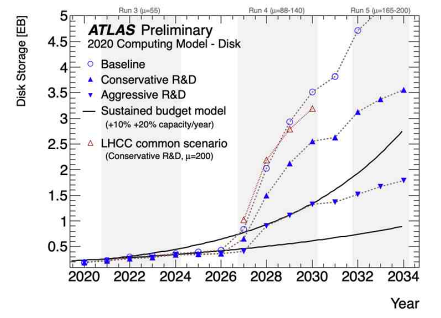
Remaining DAODs:
Significantly reduced number of additional DAOD types (10-20)

AODs:
Larger fraction only available on TAPE

ATLAS disk space status and projections

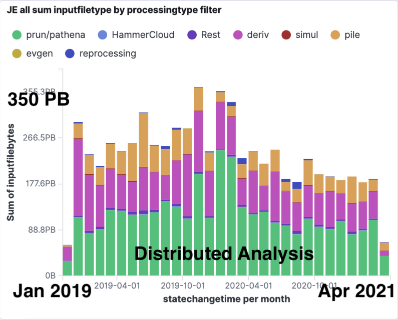


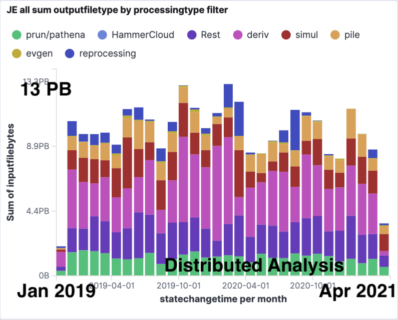
- DISK: 264 PB, filled mainly with Analysis formats (AOD/DAOD)
- Only 1-2 replicas possible because of large sample sizes
- In addition TAPE pledge of 330 PB



- **Run3:** within "flat budget"
- **Run4:** challenging to stay within "flat budget"

Processing input/output volumes PanDA in past ≈2 years





- Grid **input** processing volume ≈250 PB/month - 30-50% (≈100 PB/month) for analysis
- Grid **output** volume ≈10 PB/month - ≈2 PB/month for analysis
- Tier0 batch is not included here
- Distributed analysis users have relatively large freedom in workflow choices
- DAOD datasets largely distributed across Tier0/1/2 sites
- Extrapolations:
  - Run3: expect slightly higher numbers: more events, less formats
  - Run4: much more events should be balanced by smaller formats

HL-LHC analysis R&D scenarios

- **Baseline**: new data formats foreseen for Run 3, AthenaMT, but otherwise continues in largely the same way as in Run 2.
- **Conservative R&D**: R&D for Run3 successful: data carousel, fast track reconstruction, lossy compression, most of detector simulation with fast simulation
- **Aggressive R&D**: New developments with very significantly improve the speed or storage volumes. For analysis almost universal adoption by the physics groups of DAOD_PHYSLITE. Faster full and fast simulation, porting of code to GPUs

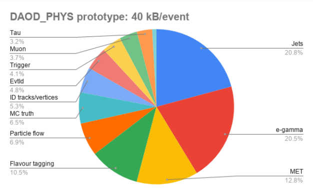
DAOD_PHYS:
Available and under commissioning for Run3

DAOD_PHYSLITE:
Advanced prototype available and more work w.r.t. systematic handling needed towards Run3

Lossy compression:
Under commissioning for Run3 in DAOD_PHYS

Containers:
Analysis and production releases and user container in production, can be used e.g. on PanDA or in local cluster

Data carousel:
In production - popular AODs and HITS kept in a disk buffer, and others staged from TAPE on-demand



Very simple HL-LHC extrapolation for events and disk

	MC			Data		
	AOD	DAOD PHYS	DAOD PHYSLITE	AOD	DAOD PHYS	DAOD PHYSLITE
events / year	$2 \cdot 10^{11}$	$2 \cdot 10^{11}$	$2 \cdot 10^{11}$	$7 \cdot 10^{10}$	$7 \cdot 10^{10}$	$7 \cdot 10^{10}$
size/event [kB]	1000	50	10	700	50	10
disk [PB/year]	200	10	2	49.0	3.5	0.7

Assumptions:

- no extra versions & no replication - this will increase the volume by a factor 2-4
- More disk space is needed for additional DAOD flavours for combined performance groups and special physics analysis
- Average size/event and no pile-up dependence assumed here

$\rightarrow$ More DAOD_PHYSLITE and less DAOD usage, AOD with tape carousel will reduce disk capacity needs

Some Computing model aspects

Analysis pipelines	For production: derivation production Individual analysis: requires proper environment like e.g. Analysis facility Analysis Preservation/RECAST with REANA is already a requirement for e.g. BSM analysis
Analysis diversity	Expect 70-80% can use DAOD_PHYS/LITE but special needs for extra formats/samples in e.g. b-physics or Combined Performance groups
Programming models	Declarative vs. procedural: support both Automating systematics, calibrations: central code for both are used in all individual analysis and is planned to be used in DAOD_PHYSLITE production how to best handle systematics in central production is under discussion and requires R&D

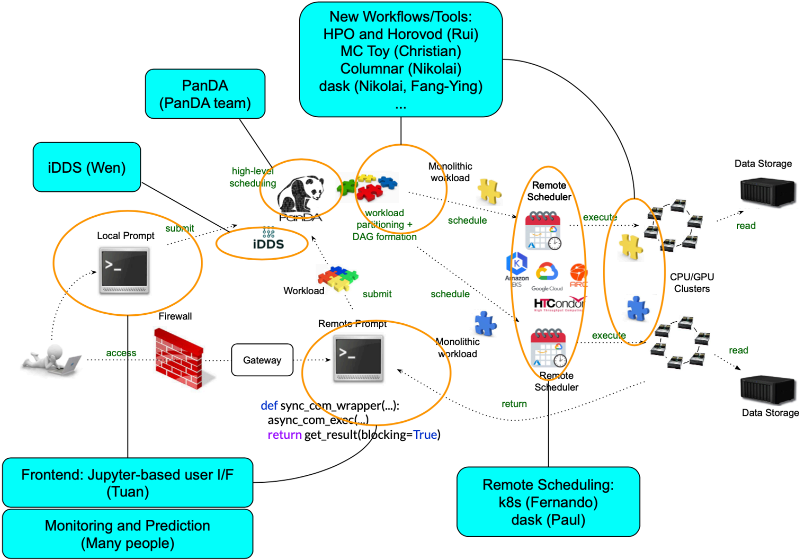
Impact of R&Ds

GPU	mainly projected to be used in production (Simul and Reco) and ML for analysis - but are there new trends ?
Analysis facilities	Active R&D within WLCG/DOMA on-going Presentation with ADC plans will be presented at vCHEP Some prototypes at small scale available e.g. in the US, Cloud R&D on-going
Jupyter Notebooks	Ideal for code prototyping but requires seamless integration with a data facility in case of bulk processing needs Unclear how many users can be served simultaneously
Trends	ROOT remains a corner stone of analysis Python analysis ecosystem (see below) Columnar data format for analysis: R&D on-going Real-time analysis R&D on-going, but offline analysis still needed
Python ecosystem	ROOT is one pilar of analysis But also support for data science tools inside/outside of HEP Important for AI/ML tools and data formats Training of new students
Languages	C++, Python

Latency

- The most crucial issue
  - Pointless if there is days of latency
- Major contributions
  - Task and job creation time
    - Performance improvement of iDDS
    - Dedicated JEDI and Panda server nodes
    - Express share for analysis
    - Shorter daemon cycles
  - Queueing time in remote schedulers
    - Dedicated computing resources, preemption, on-demand cluster spin-up, …
  - Tail in task completion time
  - Stage-in/data-ingestion and stage-out time
- User interface, monitoring, and prediction on the completion time can mitigate the issue
  - User interface: to hide asynchronous gotcha from the user
  - Monitoring: to continuously show the progress of the processing and keep the user informed
  - Prediction: to give clear and hopeful perspective to the user

Relevant Development Activities



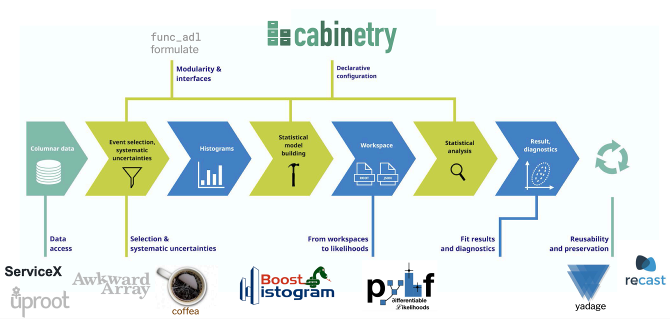
Outline

User analysis in the ATLAS experiment

IRIS-HEP

ROOT

Analysis Systems in IRIS-HEP - Focus Area Strategies

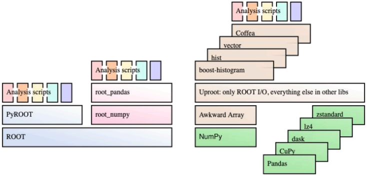


- Establish declarative specifications for analysis tasks and workflows that will enable the technical development of analysis systems to be decoupled from the user- facing semantics of physics analysis.
- Leverage and align with developments from industry and the broader scientific software community to enhance sustainability of the analysis systems.
- Develop high-throughput, low-latency systems for analysis for HEP.
- Integrate analysis capture and reuse as first class concepts and capabilities into the analysis systems.

- Analysis Grand Challenge:
    - End-to-end analysis optimization including systematics on a realistically sized HL-LHC (≈200 TB) end-user analysis dataset
    - Analysis Preservation & Reinterpretation: The ability to preserve the optimized analysis (in git repositories, docker images, workflow components, etc.), reproduce results, and reinterpret the analysis with a new signal hypothesis.

uproot

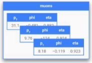
Awkward Array



- Minimalist ROOT I/O in pure Python and Numpy
- Uproot easily bridges the ROOT and the NumPy-based ecosystems
- Unlike the standard C++ ROOT implementation, Uproot is only an I/O library, primarily intended to stream data into machine learning libraries in Python.

https://github.com/scikit-hep/uproot4

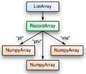
Logical view: particles as lists of nested objects




```
[[Muon(31.1, -0.481,  0.882), Muon(9.76, -0.124, 0.924), Muon(8.18, -0.119, 0.923)],
 [Muon(5.27,  1.246, -0.991)],
 [Muon(4.72, -0.207,  0.953)],
 [Muon(8.59, -1.754, -0.264), Muon(8.71,  0.185, 0.629)]
```

Physical layout: arrays grouped in a tree structure

```
offsets       0,                    3,     4,     5,       7
     pt    31.1,   9.76,   8.18,  5.27,  4.72,  8.59, 8.714
    phi  -0.481, -0.123, -0.119, 1.246, -0.207, -1.754, 0.185
    eta   0.882,  0.924,  0.923, -0.991,  0.953, -0.264, 0.629
```

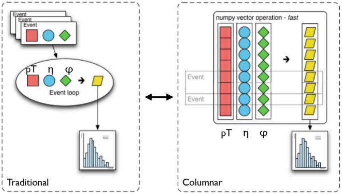


- Pure Python+Numpy library for manipulating complex data structures even if they
  - Contain variable-length lists (jagged/ragged)
  - Are deeply nested (record structure)
  - Have different data types in the same list (heterogeneous)
  - Are not contiguous in memory

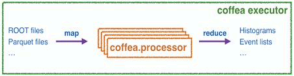
https://github.com/scikit-hep/awkward-1.0

Coffea Analysis Framework

Fermilab



New columnar data analysis concepts!



User just needs to define a high-level wrapper around user analysis code: **the coffea processor** and coffea framework will take care of everything incl. ***scaling-out***

DASK

APACHE Spark

Parsl

Distributed executors!

Smith, Nicholas, Lindsey Gray, Matteo Cremonesi, Bo Jayatilaka, Oliver Gutsche, Allison Hall, Kevin Pedro et al. "COFFEA Columnar Object Framework For Effective Analysis." https://doi.org/10.1051/epjconf/202024506012
EPJ Web of Conferences, vol. 245, p. 06012. EDP Sciences, 2020.

https://github.com/CoffeaTeam/coffea

ServiceX service

ServiceX is a scalable HEP event data extraction, transformation and delivery system (it provides user level ntuple production)

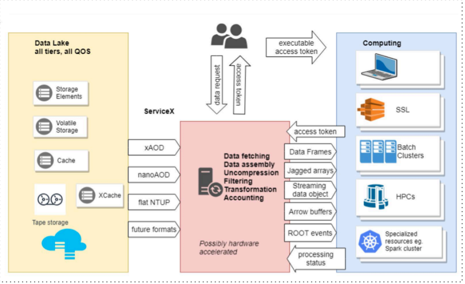
- Converts experiment-specific datasets to columns: **ATLAS xAOD/DAOD, CMS NanoAOD, ROOT Flat Ntuple**
- Enable simple cuts or simple derived columns and fields
- *Delivery:* deliver to a user or stream into Analysis System
- *Scalable:* runs on any Kubernetes cluster, scales up workers when necessary

[ServiceX doc](#)

https://github.com/ssl-hep

["Towards Real-World Applications of ServiceX, an Analysis Data Transformation System" Kyungeon Choi vCHEP](#)

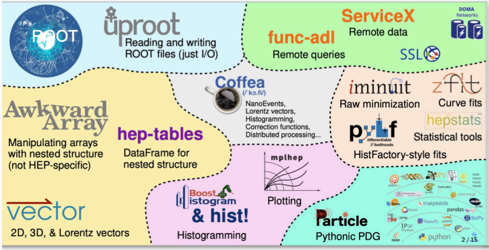
ServiceX

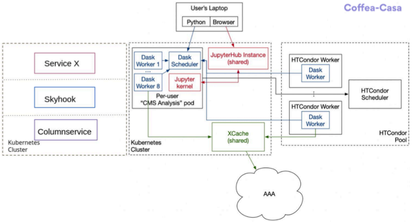


***Servicex** tested with* DASK *and*

Can be easily deployed on the local computer as well in analysis facility!

Broader ecosystem: analysis facility services development





Coffea-Casa

....and many more

(IRIS-HEP Analysis Systems)

Coffea-casa: an analysis facility prototype', vCHEP 2021 plenary

Coffea-casa: an analysis facility prototype, M. Adamec, G. Attebury, K. Bloom, B. Bockelman, C. Lundstedt, O. Shadura and J. Thiltges, *arXiv* 2103.01871

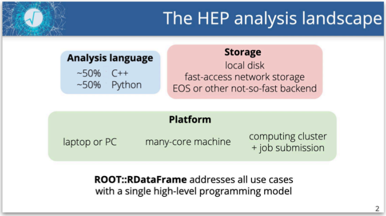
Outline

User analysis in the ATLAS experiment

IRIS-HEP

ROOT

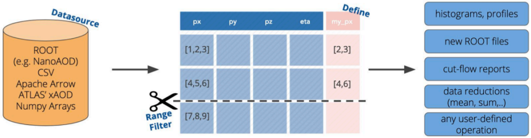
ROOT RDataframe



D Piparo, P Canal, E Guiraud et al. **"RDataFrame: Easy Parallel ROOT Analysis at 100 Threads"**

EPJ Web of Conferences **214**, 06029 (2019)
https://doi.org/10.1051/epjconf/201921406029

https://doi.org/10.1051/epjconf/201921406029




```
# enable multi-threading
ROOT.EnableImplicitMT()
df = ROOT.RDataFrame(dataset)
```

```
df = df.Range(2)
       .Define("my_px", "px[eta > 0]")
```

```
# filled in a single loop
h1 = df.Histo1D("my_px", "w")
h2 = df.Histo1D("px", "w")
```

The introduction of elements of **declarative programming** in the design (users say *what* they need to compute, RDataFrame chooses *how* to compute it) provides user-visible advantages such as less typing, increased readability and abstraction of complex operations.

RDataframe tutorials

ROOT RNTuple

Seamless transition from TTree to RNTuple

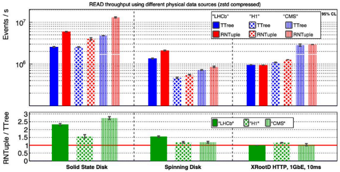
Event iteration
Reading and writing in event loops and through `RDataFrame`
`RNTupleDataSource`, `RNTupleView`, `RNTupleReader/Writer`

Logical layer / C++ objects
Mapping of C++ types onto columns
e.g. `std::vector<float>` $\mapsto$ index column and a value column
`RField`, `RNTupleModel`, `REntry`

Primitives layer / simple types
"Columns" containing elements of fundamental types (`float`, `int`, ...)
grouped into (compressed) pages and clusters
`RColumn`, `RColumnElement`, `RPage`

Storage layer / byte ranges
`RPageStorage`, `RCluster`, `RNTupleDescriptor`

RNTuple provide modular storage layer that supports files as data containers but also file-less systems (object stores)!



RNTuple R&D aiming at a **leap in data throughput**

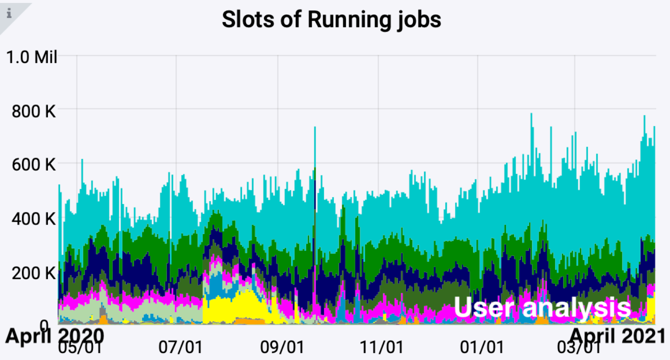
- Updated (backwards incompatible) data format for next-generation event I/O
- *Expect ~10-15% smaller files,*
  x2-5 better single-core throughput on SSD
- Aims at using modern I/O devices to the full capacity
- Modern, robust API (e.g., thread-friendly, systematic use of exceptions)

Blomer, Jakob, et al. **"Evolution of the ROOT Tree I/O."** *EPJ Web of Conferences*. Vol. 245. EDP Sciences, 2020. https://doi.org/10.1051/epjconf/202024502030

- Key components for HL-LHC:
    - Multi-step data reduction
    - Compact data format(s)
    - Calibrations and smart systematic handling
    - Smart integration of ML and emerging technologies like dedicated facilities or new data handling

BACKUP

CPU usage



- CPU pledge of 3125 kHS06
- 10-20% of analysis share on the Grid/Cloud - not HPC - mainly single core serial processing payloads
- Very diverse inputs and processing payloads in analysis
- In addition lots of final analysis happens on local batch farm or computers on individual ntuples

ATLAS DISTRIBUTED COMPUTING OVERVIEW

The ATLAS distributed computing system is centered around:

- **Workflow management system**: PanDA
- **Data management system**: Rucio
- Many **additional components**: AGIS, ProdSys, Analytics, ...
- **Resources**: WLCG grid sites, Tier0, HPCs, Boinc, Cloud
- **Shifters**: Grid, Expert and Analysis (ADCoS, CRC, DAST)