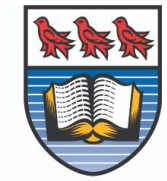
ATLAS Analysis: Current model and future plans

ROOT Users Workshop, 9 - 12 May 2022

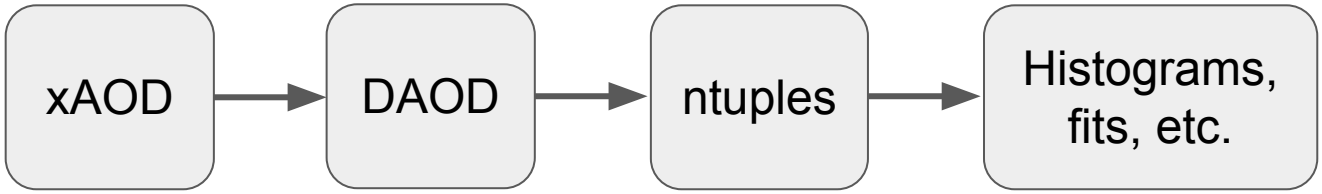
Heather Russell



University of Victoria

Overview

Oversimplified analysis flow:



Event **D**ata **M**odel (EDM): format in xAOD, DAOD files - not a simple ntuple, allows for very efficient storage of objects via decorations, shallow copies

ntuples created in dedicated analysis releases, by each (group of) analyses:

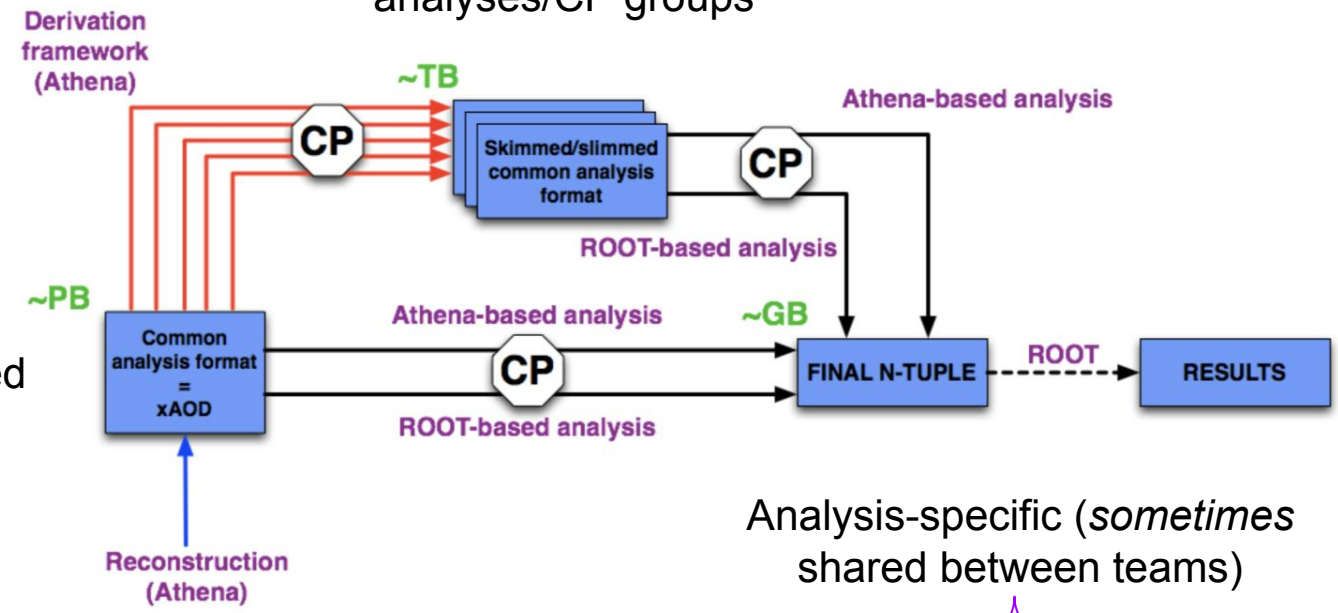
- select events
- run “CP” tools to calibrate, identify, and apply MC→data scale factors to objects
- determine **systematic variations** and save these to event weights or trees

CP tools have EDM-based interfaces (inputs are the xAOD objects)

Final step is analysis-dependent

Run 2 analysis model

~200 formats, many versions.
created on request by
analyses/CP groups



Centrally produced
~once

Analysis-specific (*sometimes* shared between teams)

ntuple-making

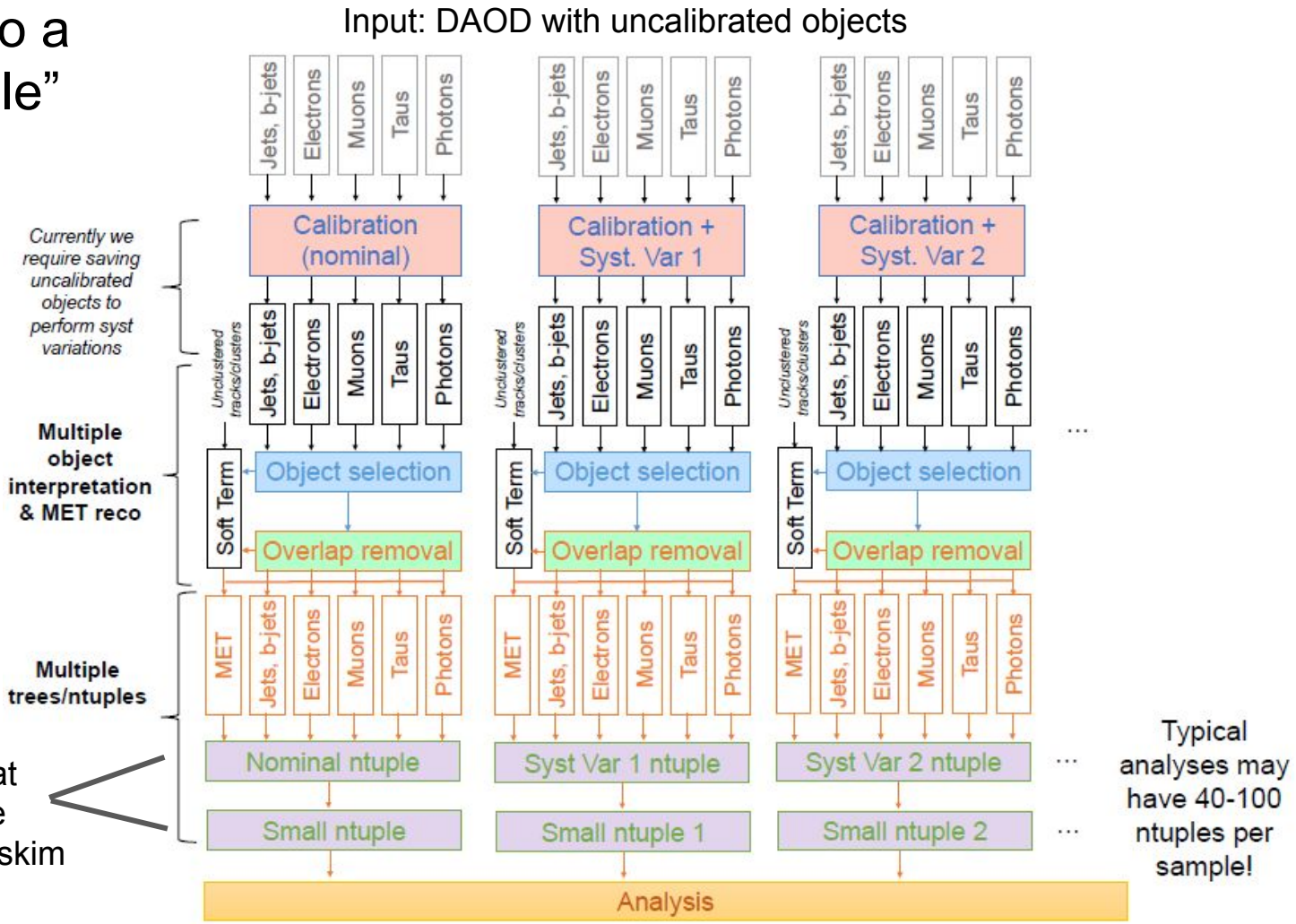
~a dozen shared frameworks, plus many more custom ones
Lots of duplicated work on implementation of all the configurations of CP tools (calibrations + systematic uncertainties)

ntuple-analysing

Mostly analysis-specific, using a wide range of tools from simple ROOT macros to RDataFrame to uproot, etc.

Run 2 analysis model

The path to a "final ntuple"



With systematic variations, analyses have O(1 - 10 TB) of data to process for their analysis

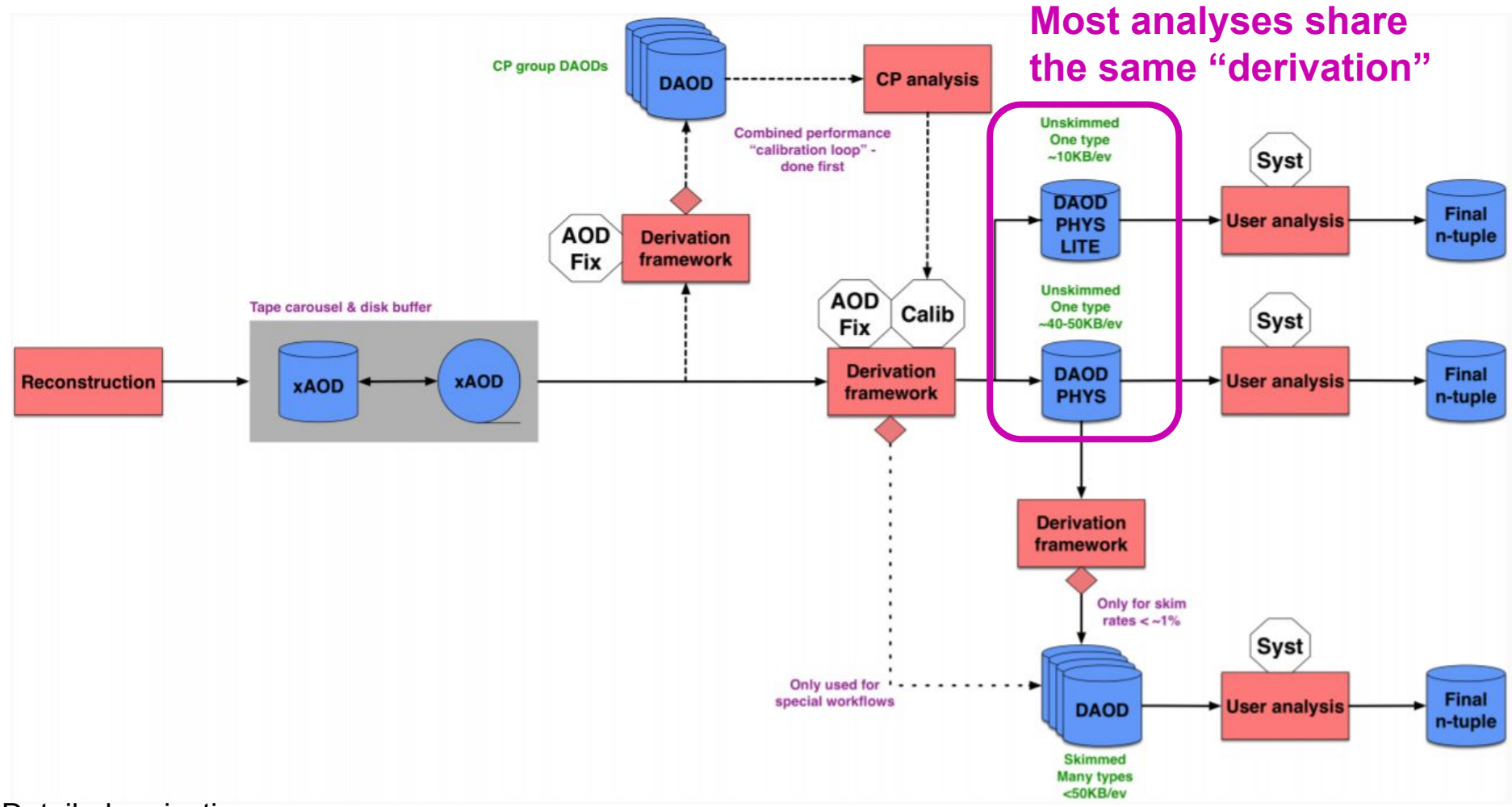
Targeted improvements

(1) Make analysers' lives easier
(2) Reduce CPU usage
(3) Minimize disk space

Many things users want jibe with our technical requirements:

- Not over-storing data or MC [efficient skimming, efficient sharing]
- Fast processing of events [efficient organization of algorithms and selection]
- Practical best practices [no long grid wait times for very fast jobs, no large # of users running long jobs on lxplus nodes. Easiest workflow should be the recommended one]
- Efficient use of analysers' time [minimize bookkeeping, babysitting of grid jobs, manual repetition of automizable tasks]
- Optimization of systematic uncertainties [balance UX & "correctness"]
- **Usable by non-experts [*i.e.* students with little programming experience]**

Run 3 analysis model - Overview



Detailed projections:
https://cds.cern.ch/record/2696416/files/ATL-SOFT-SLIDE-2019-810.pdf

Run 3 analysis model - Details

xAOD Type	Size per event
AOD	600 kB
DAOD	40 – 450 kB
DAOD_PHYS	50 kB
DAOD_PHYSLITE	10 kB

Skimmed events: only for special cases in Run 3

Unskimmed, uncalibrated events: Main Run 3 format

Unskimmed, calibrated events: available Run 3, main format by HL-LHC

PHYS slots into existing analyses with small modifications for new software

⇒ PHYSLITE is the fun part!

NB: Not everyone will be able to use PHYSLITE; we aim for 80% of analyses

PHYSLITE uses xAOD EDM, and will likely be based on **RNtuple**

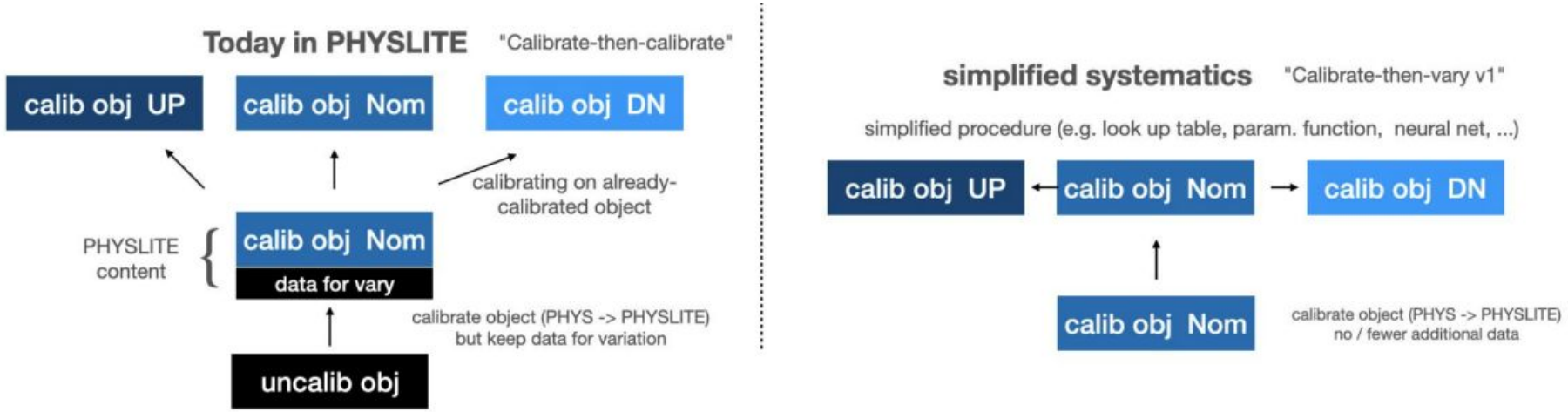
Skimmed PHYSLITE should replace big ntuples many groups save

Big question: can we process xAODs and run CP tools using RDataSource/RDataFrame? How far can we push this up our workflow?

Improving the analyser experience

(1) Calibrations are run in the **AOD → PHYSLITE** step: PHYSLITE contains calibrated objects + info for syst. variations
   (a) Fewer tools needed for analysis jobs (faster, simpler, less error-prone)

(2) Investigating simplification of CP tools (e.g. by parametrizing all systematic uncertainties) for PHYSLITE. *Could* work with RDF.vary() but will require significant R&D
   (a) Vastly reduced storage, faster, simpler. Problematic for e.g. overlap removal and MET.



(3) Interactive PHYSLITE via **analysis facilities**. Example target workflow: Jupyter hub interface → batch/Dask/etc → plot

Solves problem of how to process O(100 TB) of HL-LHC PHYSLITE

Statistical tools

HistFactory forms the backbone of many of ATLAS's statistical tools for both measurements and searches

⇒ we are always interested in developments!

pyhf is also increasingly used within ATLAS

- partly because it introduced JSON configuration
  - more easily readable format than xml, and a nice companion to JSON-formatted HEPData
  - Easing combinations and publishing likelihoods (e.g. https://atlas.cern/updates/news/new-open-likelihoods)
- Very interested in the new RooFit JSON developments (see talk from C. Burgard - https://indico.fnal.gov/event/23628/contributions/240368/);
  - would be excellent if the pyhf JSON configuration were compatible with this

Summary + thoughts

We should not assume that everyone will perform their full analysis chain using ROOT

- Desirable: ROOT integrating seamlessly with the scientific python ecosystem
- Concrete example: ONNX format (facilitates interoperability of machine learning software) with SOFIE

Suggested workflows, best practices also need to be easy for non-experts to understand

RDataSource and RDataFrame are allowing us to investigate pushing columnar analysis higher up the production chain

Processing HL-LHC amounts of data/MC will require distributed analysis workflows: prototyping these now