Systematics

Stephan Hageböck

23 May 2022

Why should we care?

- Can we make the life of the analysers easier?
- Can we reduce the amount of work they have to repeat?
- Can we help to compute efficiently?
- How does smart handling affect the space occupied by NTuples?

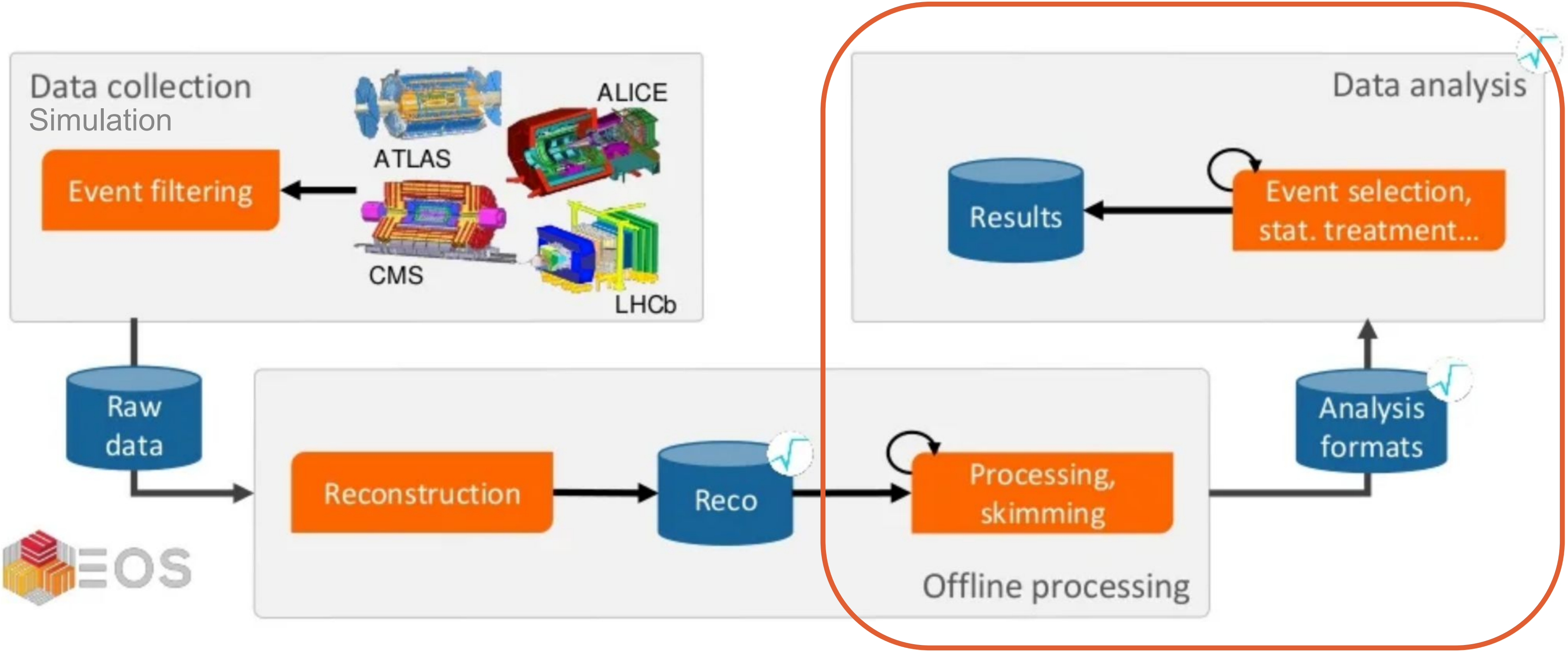
"I'd rather talk about anything than systematics; unless I have to."

"I don't like them. But if someone could get them out of the way for me, that would be great."

"We leave it up to the analysis groups."

"There's central recommendations."

Typical data pipeline



Types of Systematics

Weight Systematics

- Changes the likelihood to observe an event
- Typically for efficiencies
- All other (event) quantities unaffected

Fast to evaluate if done well
~ 150*

Object Systematics

- Changes quantities in an event
- Particle momentum / energy
- Need to rerun selection and plotting steps

More effort to evaluate
~ 30*

The Rest

- Doesn't fit into this scheme
- Specific to an analysis
- Probably not our business?

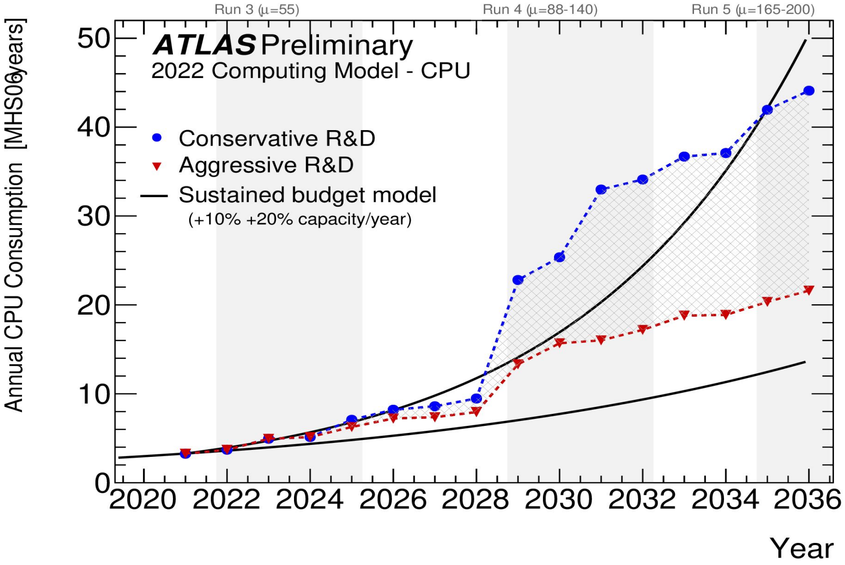
?

* (*ATLAS H→bb biased*)

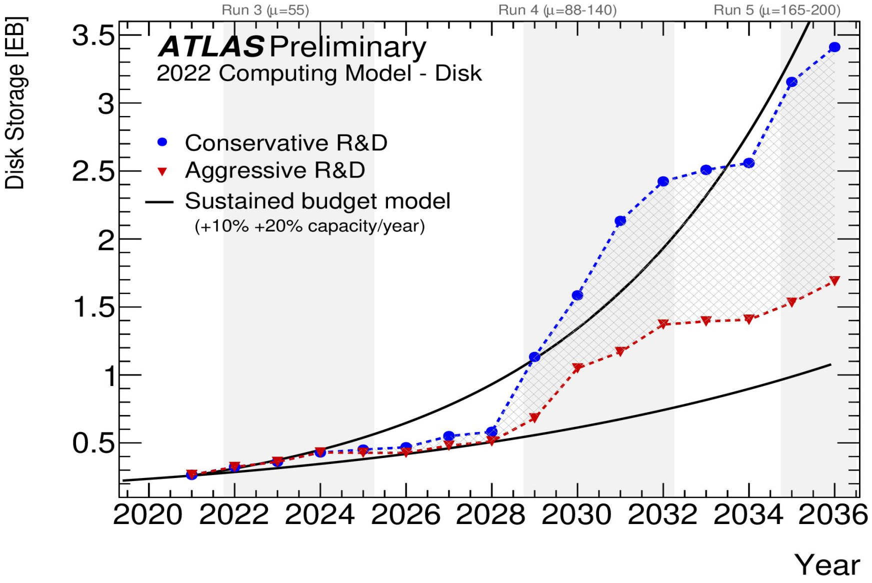
Are Systematics a Part of the Problem?

Graeme this morning: "**We have to make it cheaper per event**"

CPUs

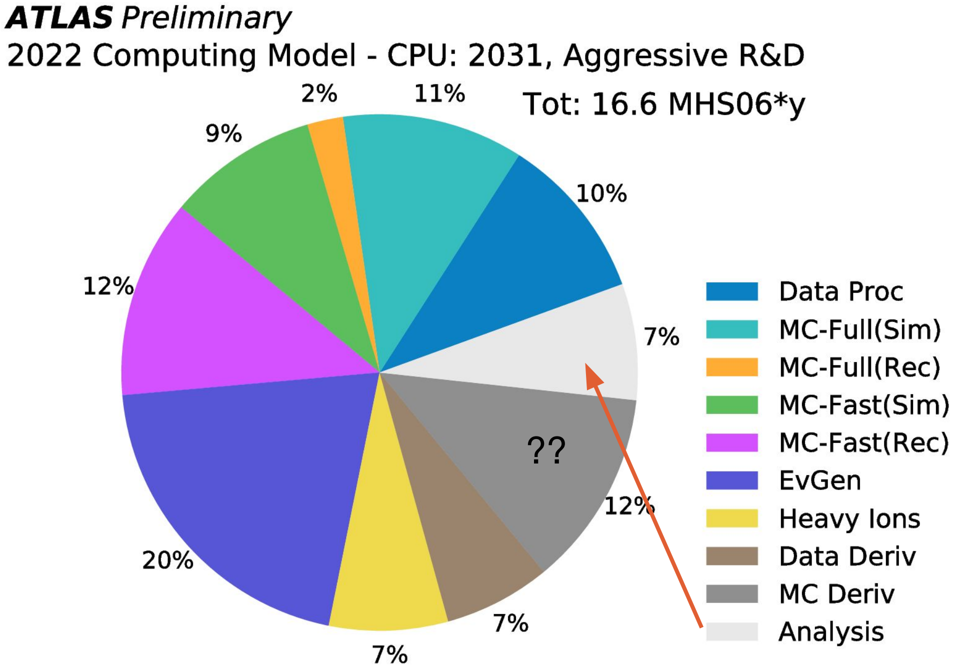


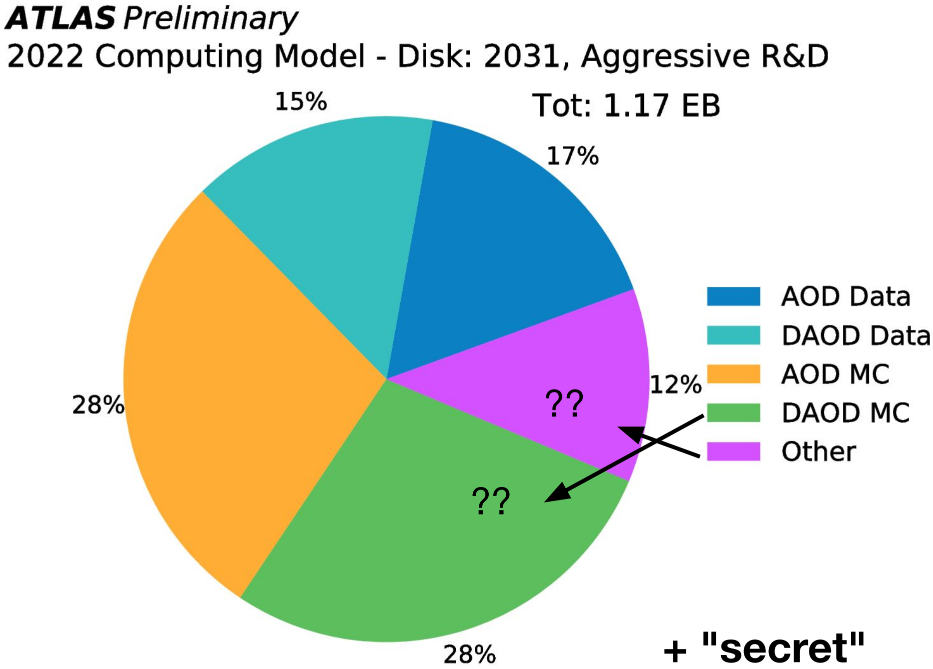
Disks



Source

Are Systematics a Part of the Problem? → Yes!





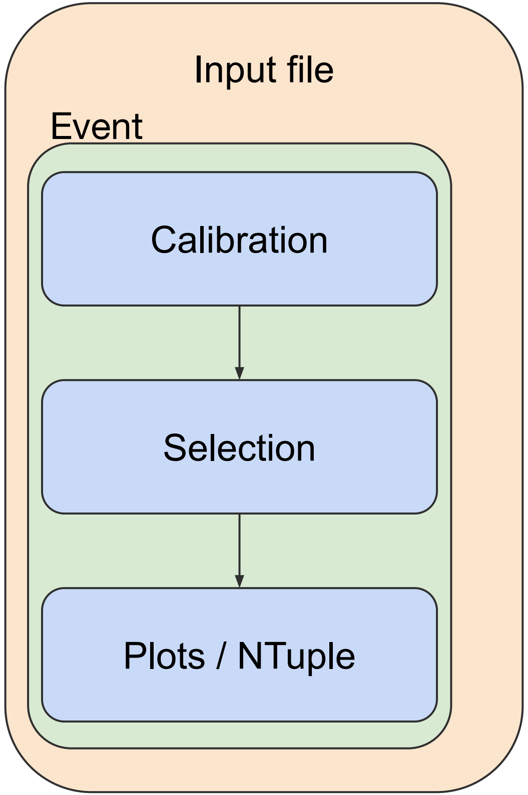
+ "secret" computing

Source

The Classic Way to Evaluate Systematics

```
for event in events:
    calibration(event, toolList, ...)
    selection(event, toolList, ...)
    plots(event, ...)
```

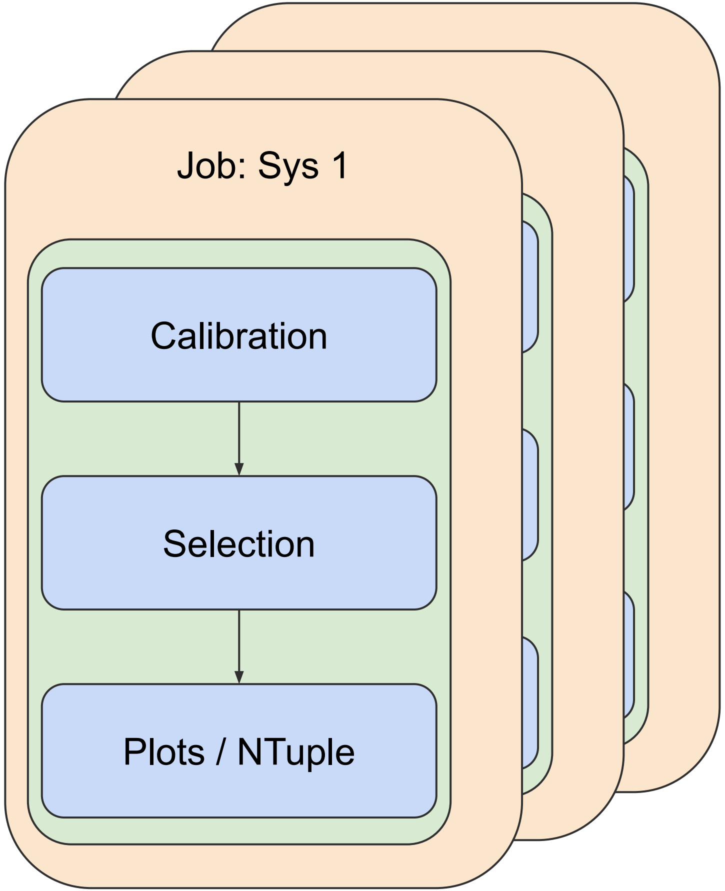
- The "easiest" is to rerun everything with a variation switched on
- Some tools encourage this



How not to do it

```
toolList["MuonCalibration"].
  applySystematicVariation(sys)

for event in events:
    calibration(event, toolList, ...)
    selection(event, toolList, ...)
    plots(event, ...)
```

- The "easiest" is to rerun everything with a variation switched on
- Wastes CPU cycles and space
- Sometimes no way around

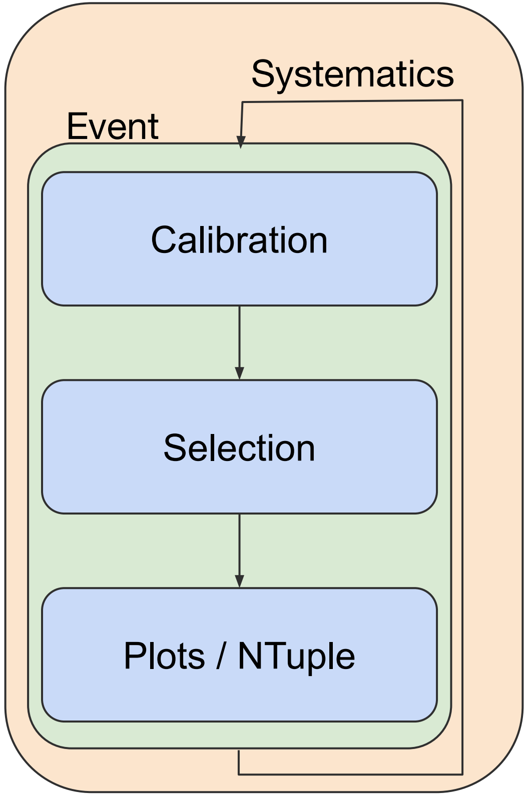


See much more details in
[Data Analysis WG meeting](#)
[ROOT WS: Analysis Model](#)

Better?

```
for event in events:
  for sys in systematicsList:
    for tool in toolList:
      tool.applySystematicVariation(sys)

    calibration(event, toolList, ...)
    selection(event, toolList, ...)
    plots(event, ...)
```

- Experiment provides common tools / interfaces for systematics
- Extra loop in every event
- Cycle through all variations
- Wasteful for weight systematics



Save one NTuple for each run??

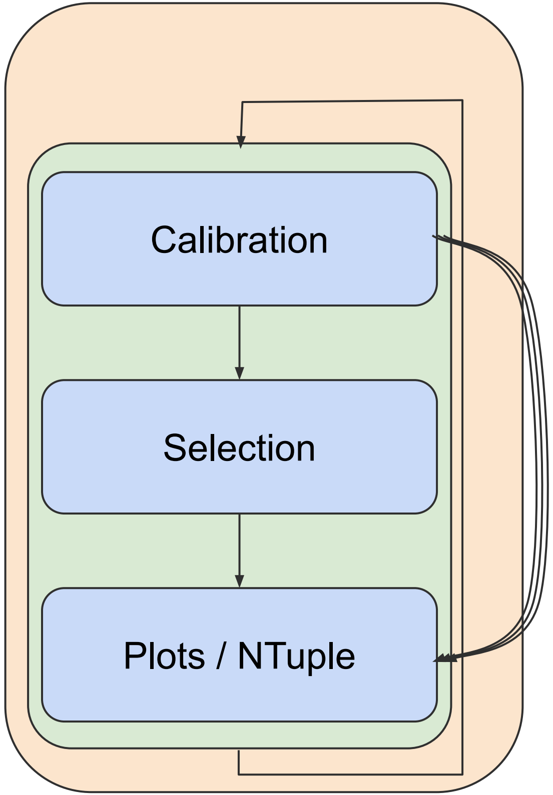
See much more details in
[Data Analysis WG meeting](#)
[ROOT WS: Analysis Model](#)

Better ??

```
for event in events
  for sys in objectSystematics:
    for tool in toolList:
      tool.applySystematicVariation(sys)

    calibration(event, toolList, ...)
    selection(event, toolList, ...)
    plots(event, ...)
```

Weights

- Experiment provides common tools / interfaces for systematics
- Only rerun for object systematics
- Transfer weights from calibration to plotting stage
- Can you do this thread safe?



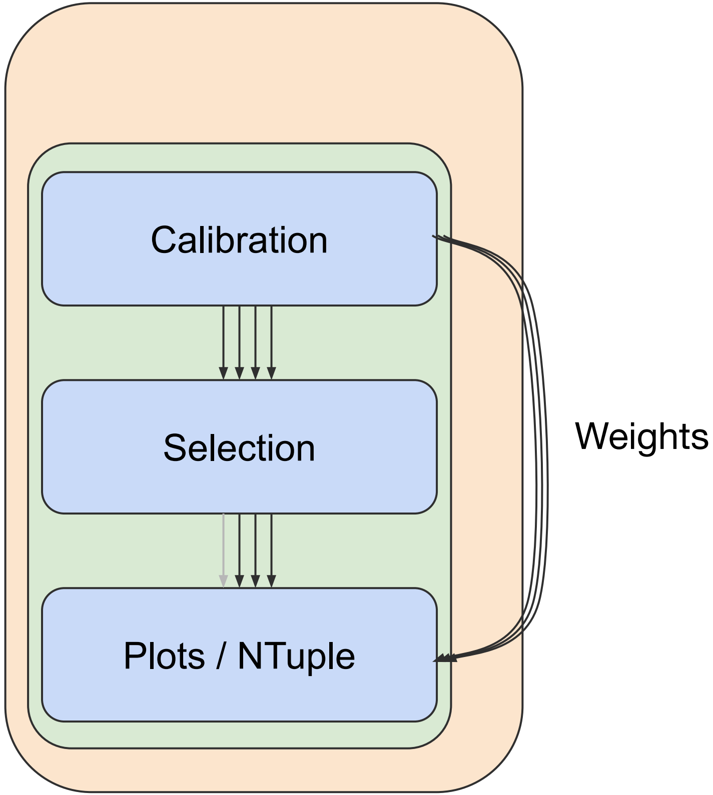
O(10) - O(100) speed up have been observed

See much more details in
[Data Analysis WG meeting]
[ROOT WS: Analysis Model]

Best ???

```
df_calib = calibration(df_input, tools, ...)
df_calib = variations(df_calib, tools, ...)
df_select = selection(df_calib, toolList, ...)
plots(df_select, ...)
```

- Experiments provide common tools / interfaces for systematics
- Analysers say what, frameworks know how (both weight/object systematics)
- Only (re-)run necessary selection / plotting steps

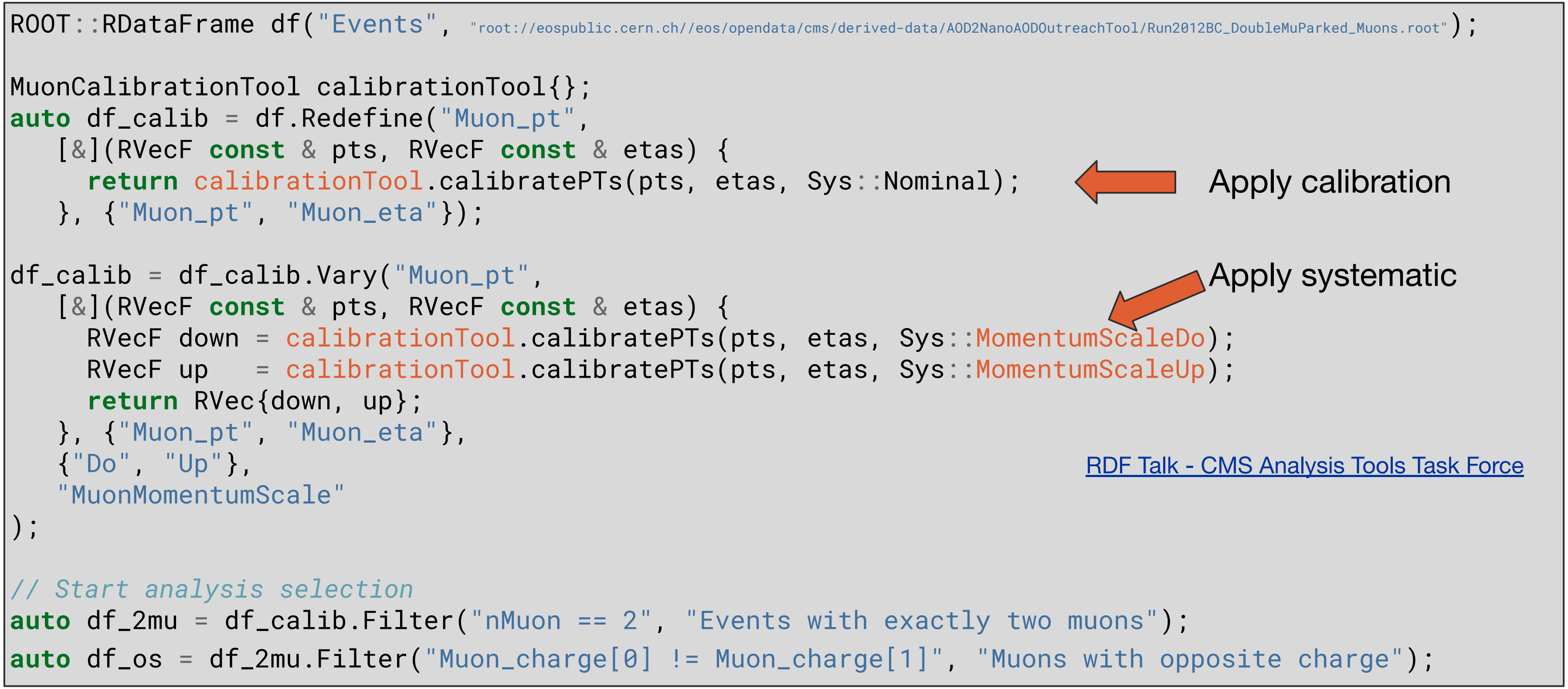


Core Software to the Rescue? – The Present

```
ROOT::RDataFrame df("Events", "root://eospublic.cern.ch//eos/opendata/cms/derived-data/AOD2NanoAODOutreachTool/Run2012BC_DoubleMuParked_Muons.root");

MuonCalibrationTool calibrationTool{};
auto df_calib = df.Redefine("Muon_pt",
   [&](RVecF const & pts, RVecF const & etas) {
     return calibrationTool.calibratePTs(pts, etas, Sys::Nominal);
   }, {"Muon_pt", "Muon_eta"});









// Start analysis selection
auto df_2mu = df_calib.Filter("nMuon == 2", "Events with exactly two muons");
auto df_os = df_2mu.Filter("Muon_charge[0] != Muon_charge[1]", "Muons with opposite charge");
```

Apply calibration

Core Software to the Rescue? – The Future



```
ROOT::RDataFrame df("Events", "root://eospublic.cern.ch//eos/opendata/cms/derived-data/AOD2NanoAODOutreachTool/Run2012BC_DoubleMuParked_Muons.root");

MuonCalibrationTool calibrationTool{};
auto df_calib = df.Redefine("Muon_pt",
   [&](RVecF const & pts, RVecF const & etas) {
     return calibrationTool.calibratePTs(pts, etas, Sys::Nominal);
   }, {"Muon_pt", "Muon_eta"});

df_calib = df_calib.Vary("Muon_pt",
   [&](RVecF const & pts, RVecF const & etas) {
     RVecF down = calibrationTool.calibratePTs(pts, etas, Sys::MomentumScaleDo);
     RVecF up   = calibrationTool.calibratePTs(pts, etas, Sys::MomentumScaleUp);
     return RVec{down, up};
   }, {"Muon_pt", "Muon_eta"},
   {"Do", "Up"},
   "MuonMomentumScale"
);

// Start analysis selection
auto df_2mu = df_calib.Filter("nMuon == 2", "Events with exactly two muons");
auto df_os = df_2mu.Filter("Muon_charge[0] != Muon_charge[1]", "Muons with opposite charge");
```

Apply calibration

Apply systematic

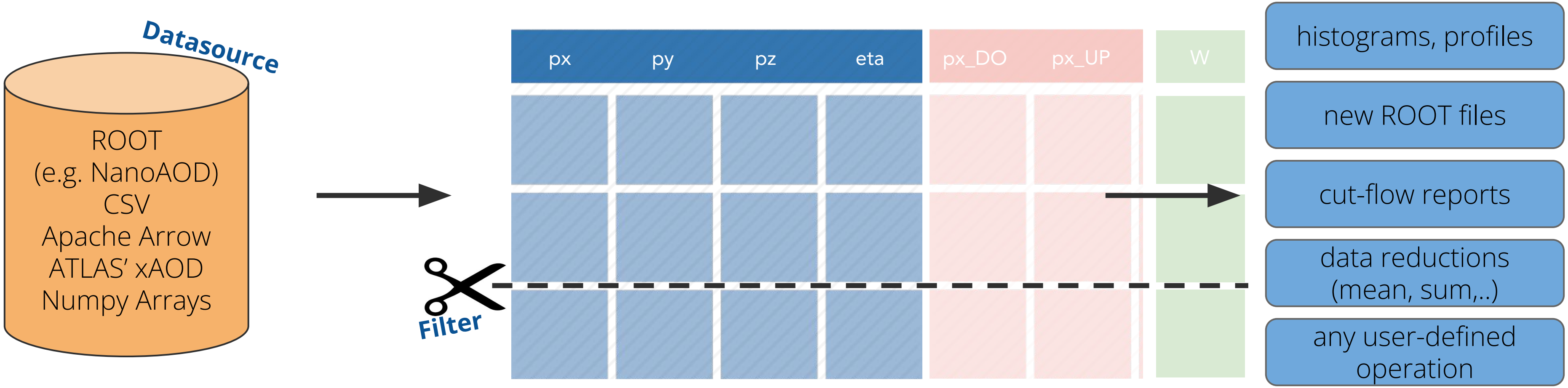
RDF Talk - CMS Analysis Tools Task Force

Core Software to the Rescue? – The Vision

```
ROOT::RDataFrame df("Events", "root://eospublic.cern.ch//eos/opendata/cms/derived-data/AOD2NanoAODOutreachTool/Run2012BC_DoubleMuParked_Muons.root");

auto df_calib = MyExperiment::ApplyStandardCalibrationAndSystematics(df);

// Start analysis selection
auto df_2mu = df_calib.Filter("nMuon == 2", "Events with exactly two muons");
auto df_os = df_2mu.Filter("Muon_charge[0] != Muon_charge[1]", "Muons with opposite charge");
```

- Both ROOT's RDF and Coffea are improving support for weight and object systematics
- Analysis frameworks could leverage those to simplify work of analysers
- Requires a lot of planning and communication
- → Reduced analysis formats?

Systematics and Columnar Thinking



- Weight systematics just add columns that affect mostly histograms / reductions
- Object systematics add columns that replace others when creating a "systematic histogram" / reduction

Reduced Formats

- Might be too condensed to (re-)compute some systematics
  - Create inputs with systematic applied?
  - Write varied quantities next to nominal data? (Friend tree, decorations?)
  - Update of calibration requires regenerating reduced format
- Reduced formats might be good enough for many systematics, though!
- Should we strive for systematics that can be (re-)computed on a reduced format?
  - Fast to recompute, on the fly
  - Save disk space
  - I/O throughput + space vs. CPU tradeoff

Current status

See Data Analysis WG meeting
CMS Analysis Tools Task Force
ROOT WS: Analysis Model
See Backup for info from Belle II

- ATLAS, CMS, Belle II have put great consolidation effort into common tools with standardised interfaces
- It's often up to analysis groups to implement systematics efficiently
    - Many have figured out good solutions for weights
    - Object systematics less clear
- Frameworks like RDataFrame and coffea are improving systematics support, efficient evaluation
- Reduced formats might be a game changer (for better or worse)
- All-in-one-go systematics on nanoAOD for CMS?

Current status

- LHCb
    - "What do you mean with common systematics? Everybody is on their own" *
    - * Not quite: Particle ID is standardised arxiv:1803.00824
- Neutrinos (here: a reply from DUNE)
    - Main focus are cross-section and flux (/ beam) systematics
    - Involves a lot of generator parameter tuning, and there's ideas for it
    - Detector systematics: "All experiments are doing what they can with the limited manpower they got"
- If you know more, **we need you**

Some Points to Consider

- Systematics mean replication of work in every analysis group
- Central tool for everything: Black-box solution?
- How much CPU time / disk space are we wasting due to inefficient handling?
  → O(10) - O(100) speed differences have been observed!
- **Central maintenance: effort spent, recognition?**
  - Effort to create correct result for paper: Acceptable
  - "I want to contribute to central maintenance": No time / Won't get a paper with this / What's your next job?

- **Is everybody on their own when it comes to systematics?**
- **What should *we* do to help / streamline / unify / disseminate knowledge?**

CERN

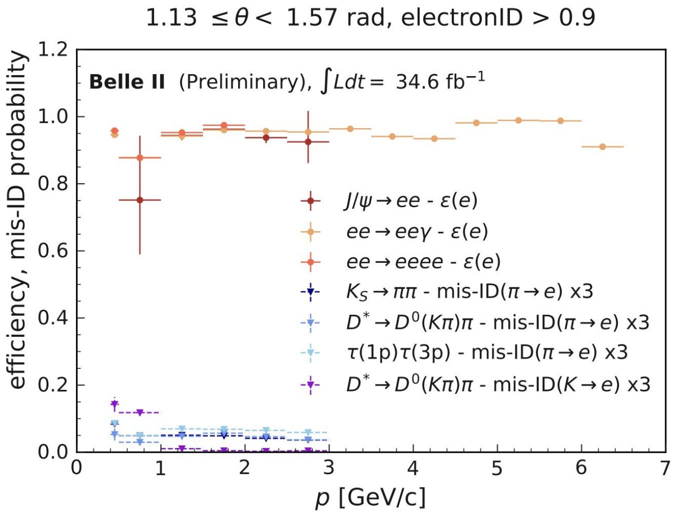
Systematics at Belle II

- Centralized efforts for determination of systematic uncertainties, organized in Physics performance subgroups.
  - **Trigger**, **Tracking & vertex**, **Neutrals, Particle-ID**, and **Beam & counting** performance**.**
  - Provide tools, documentation, and advice to analysts.
  - Regular communication with Physics, Data Production and Software groups.

- Syst uncertainties associated to detector imperfections and Data/MC correction factors are determined via control samples (cosmic data, tag-probe datasets, etc).

- Relevant payloads distributed via conditions DB.
  - ROOT objects stored as payloads in the DB.
  - Intervals of Validity (IOV) specify starting and ending experiments and runs for a given payload.
  - Transparent access via the Basf2 Software Framework.

Example on theta bin for electronID efficiencies and hadron-lepton mis-identification rates overlaid:



BELLE2-NOTE-PL-2020-027

Systematics Framework

- Designed to automate and unify the calculation of systematic uncertainties or Data/MC weights.

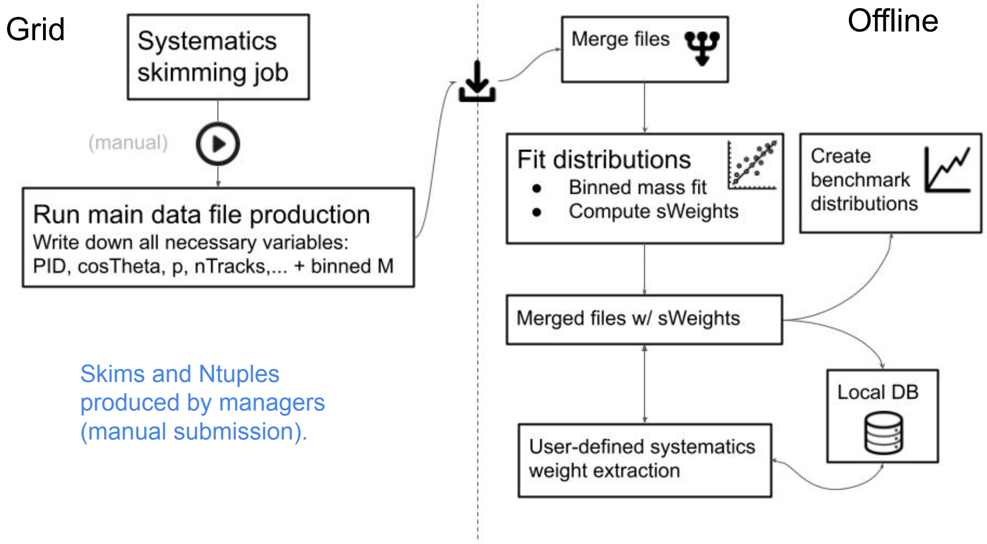
- Based on b2luigi and largely inspired by the PIDCalib framework of LHCb.

- The framework interacts mostly with control sample skim

- Weight extraction performed locally and delivered to analysts through a particle weighting module implemente in Basf2.

Use cases:

- Uncertainties associated to charged or neutral PID selection or generic MVA selection
- Continuum suppression related uncertainties, etc.
- Separation of signal from background in data is one of the key components of the framework. The best approach is to apply sPlot technique. sWeights should be added to the main ntuple with PID and other related variables.


Grid
Offline
Systematics skimming job
(manual)
Run main data file production
Write down all necessary variables:
PID, cosTheta, p, nTracks,... + binned M
Merge files
Fit distributions
Binned mass fit
Compute sWeights
Create benchmark distributions
Merged files w/ sWeights
Local DB
User-defined systematics weight extraction

Skims and Ntuples produced by managers (manual submission).

Gathering and download of files to be automated with Rucio.

SCIENTIFIC DATA MANAGEMENT
RUCIO