Analysis preservation in HEP
It matters and we can do better

(with a heavy LHC(b) bias)

UK-HEP 2024

N. Skidmore
Nov 2024

Story time…

Me: 2014 - 2017

Why do we need to preserve?

Analyses now take longer than a PhD (> 4 years)

- The analysis will have to be “handed over” to a new student (or orphaned 😞)

More complex analyses require more people

- High person-turnover is a feature of our fields
- Analysis preservation infrastructure aids collaboration

What do we need to preserve?

The physics

Analysis note

The data

nTuples

Data provenance

The code and how to run it

Git repo

Documentation

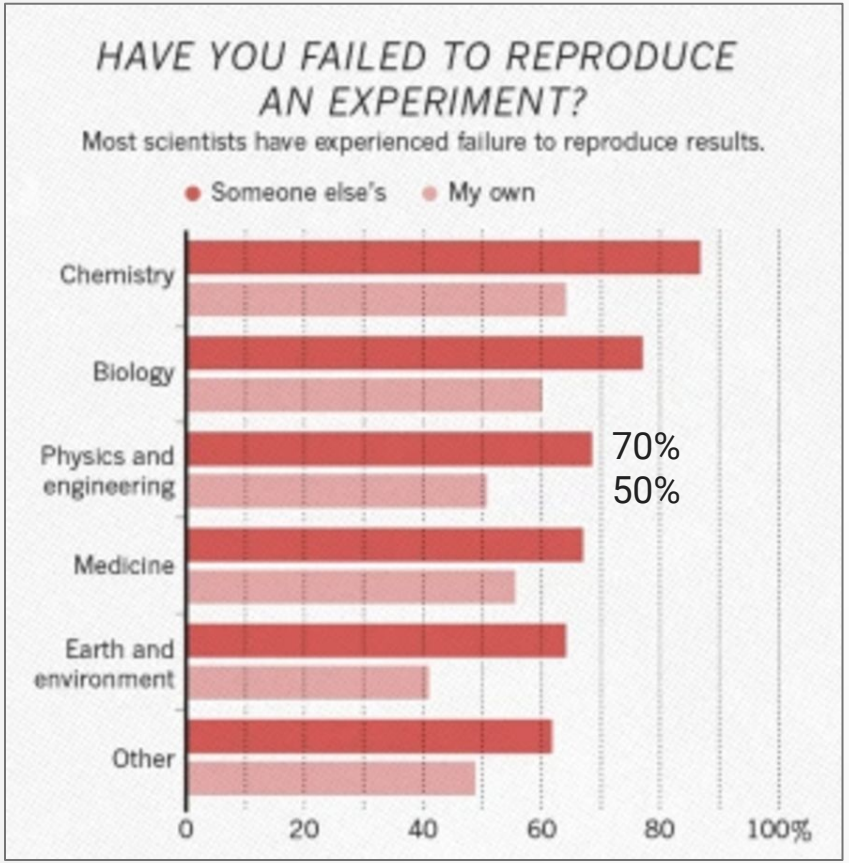
Workflow manager

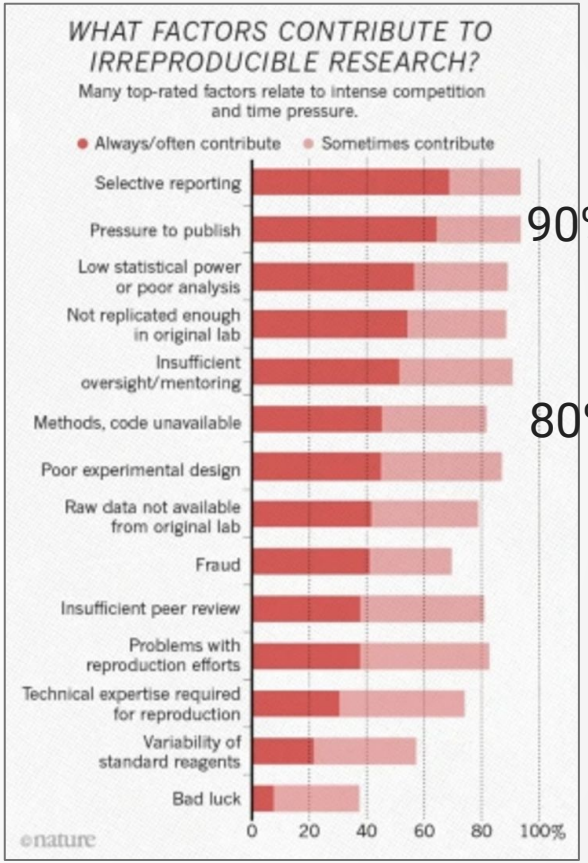
The ability to run the code

GitLab CI

Env preservation

We are bad at sharing code





www.nature.com/articles/533452a

We are bad at sharing code

Theorems, lemmas, corollaries…

Inspired by www.software.ac.uk/blog/top-ten-reasons-not-share-your-code-and-why-you-should-anyway

We are bad at sharing code

“”””The proof is too ugly to show anyone else. It would be too much work to rewrite it neatly so that others could read it.”””””

Inspired by www.software.ac.uk/blog/top-ten-reasons-not-share-your-code-and-why-you-should-anyway

We are bad at sharing code

“”””I didn't actually prove the theorem - my student did. They have since graduated and now work as a Quant.
But the student was very good… I'm sure the proof was correct.”””””

Inspired by www.software.ac.uk/blog/top-ten-reasons-not-share-your-code-and-why-you-should-anyway

We are bad at sharing code

“”””Giving the proof to my competitors would be unfair to me. It took years to prove this theorem, and the same idea can be used to prove other theorems.”””

Inspired by www.software.ac.uk/blog/top-ten-reasons-not-share-your-code-and-why-you-should-anyway

We are bad at sharing code

Table 10: Top Reasons Not to Share Code

	Not Share
The time it takes to clean up and document for release	77.78%
Dealing with questions from users about the code	51.85%
The possibility that your code may be used without citation	44.78%
The possibility of patents or other IP constraints	40.00%
Legal barriers, such as copyright	33.72%
Competitors may get an advantage	31.85%
The potential loss of future publications using this code	31.11%
The code might be used in commercial applications	28.15%
Availability of other code that might substitute for your own	21.64%
Whether you put in a large amount of work building the code	20.00%
Technical limitations, ie. webspace platform space constraints	20.00%

+ **Student left**

+ **Self -conscious about code**

dx.doi.org/10.2139/ssrn.1550193

10 rules for Reproducible Computational Research

Rule 1: For Every Result, Keep Track of How It Was Produced

Rule 2: Avoid Manual Data Manipulation Steps

Rule 3: Archive the Exact Versions of All External Programs Used

Rule 4: Version Control All Custom Scripts

Rule 5: Record All Intermediate Results, When Possible in Standardized Formats

Rule 6: Always Store Raw Data behind Plots

Rule 7: Generate Hierarchical Analysis Output

Unrealistic with big datasets

Rule 8: For Analyses Including Randomness, Record Random Seeds **Important for pheno!**

Rule 9: Connect Textual Statements to Underlying Results

Rule 10: Provide Public Access to Scripts, Runs, and Results

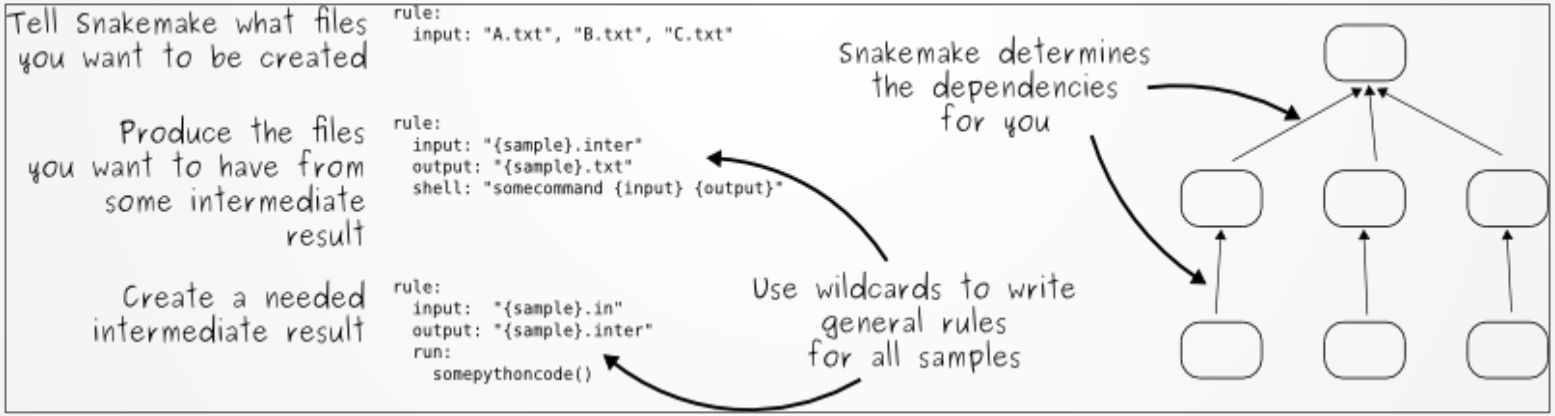
doi.org/10.1371/journal.pcbi.1003285

Rule 1: For Every Result, Keep Track of How It Was Produced

Workflow managers

- Breaks analysis into bite-size rules with input, output and command
- Preserves:
  - How to run scripts
  - How every intermediate result is produced (the workflow)
  - The dependency between analysis stages

Snakemake

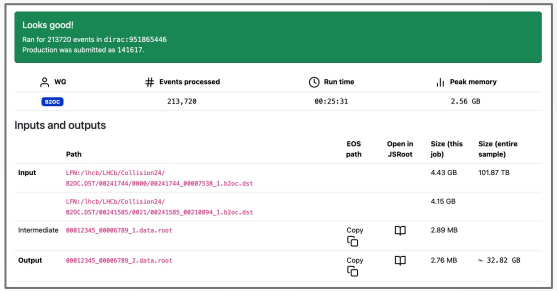


Rule 2: Avoid Manual Data Manipulation Steps

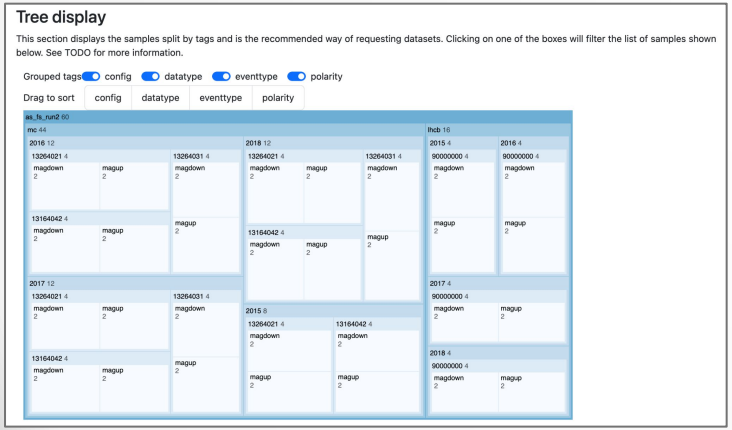
Centralise as much as possible

LHCb analysis productions for declarative ntupling

- Centralise and automate ntuple creation with DIRAC transformation system
  - Full job testing on GitLab CI
  - Full data provenance



Comprehensive job testing through GitLab pipelines



Simple yaml job configuration

```yaml
defaults:
  application: DaVinci/v64r10@x86_64_v2-el9-clang16-opt
  output: DATA.ROOT
  options:
    entrypoint: bs2dspi_run3.dv_simple:alg_config
    extra_options:
      input_raw_format: 0.5
      input_type: ROOT
      simulation: False
      data_type: "Upgrade"
      geometry_version: run3/trunk
      conditions_version: master
      input_process: "TurboPass"
      input_stream: "b2oc"
  inform:
    -
  wg: B2OC

{%- set datasets = [
 ('2024Data', 'MagDown', '24c2'),
 ('2024Data', 'MagUp', '24c2'),
```

What application to run

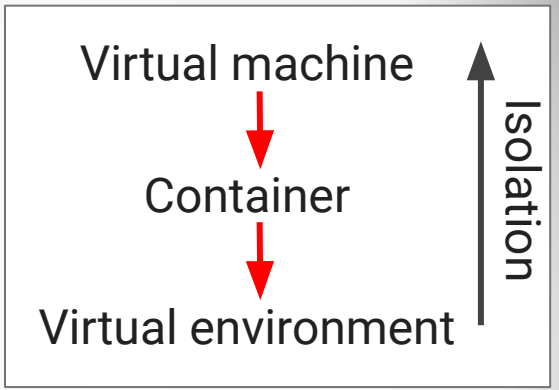
Job options

Data to run on

Rule 3 - Archive the Exact Versions of All External Programs Used

Containers and virtual environments

- Containers encapsulate a computing environment including OS
- Virtual environments only encapsulate Python dependencies



Questions for a new collaborator

1. How do I login?
2. How do I access the data?
3. How do I setup my environment?
4. How do I get the analysis s/w libraries?
5. How do I run the analysis steps?

Containers

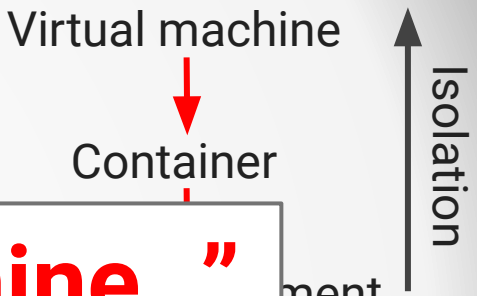
Virtual environments

These do take some time and setup but they enable

- Quick onboarding
- Efficient collaboration

Rule 3 - Archive the Exact Versions of All External Programs Used

Containers and virtual environments



- Containers encapsulate a computing environment including OS
- Virtual environments only encapsulate Python dependencies

“Well it works on my machine…”

Questions

1. How do I login?
2. How do I access the data?
3. How do I setup my environment?
4. How do I get the analysis s/w libraries?
5. How do I run the analysis steps?

Containers

Virtual environments

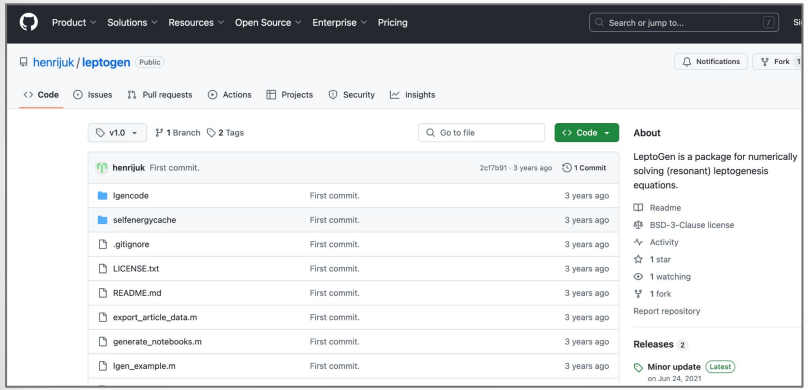
These do take some time and setup but they enable

- Quick onboarding
- Collaboration
- Publishable software
- Reuse and repurposing

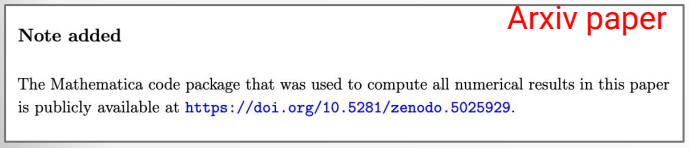
Rule 3 - Archive the Exact Versions of All External Programs Used

Package and version software

- Encourage packaging, versioning and long term maintenance of software tools



Archive the tag of a repo

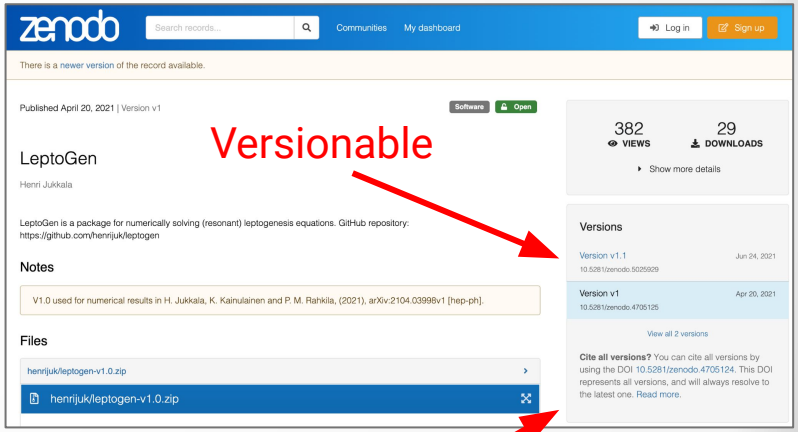


Note added

The Mathematica code package that was used to compute all numerical results in this paper is publicly available at https://doi.org/10.5281/zenodo.5025929.

Arxiv paper

zenodo.org/records/4705125



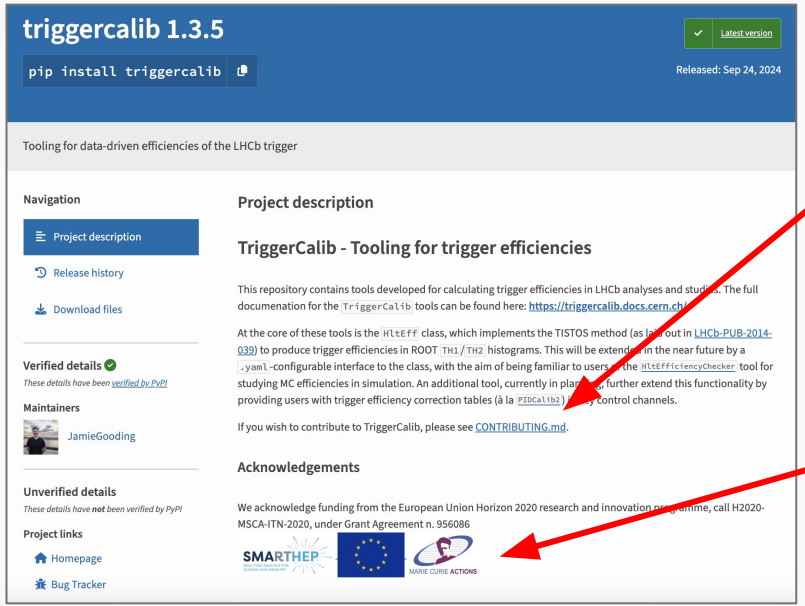
Versionable

Persist identifiers - DOIs

Rule 3 - Archive the Exact Versions of All External Programs Used

Package and version software

pypi.org/project/triggercalib/



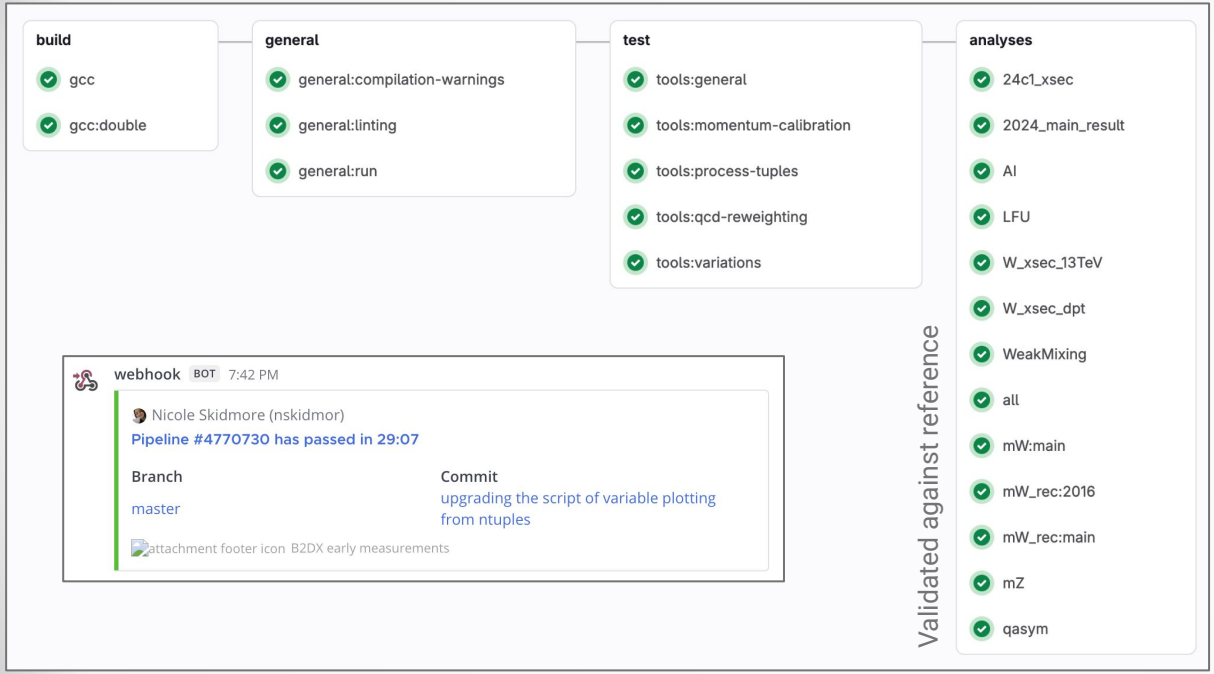
PyPI package installable with pip

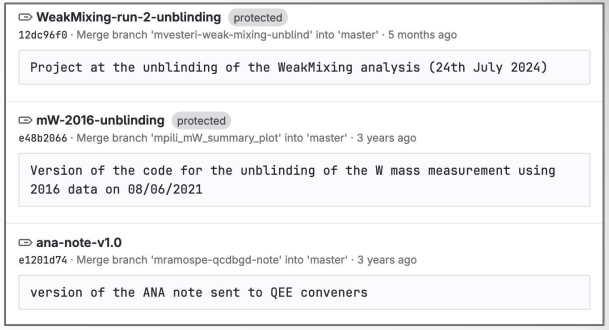
Encourage contributions

Can acknowledge funding agencies

Rule 4: Version Control All Custom Scripts

Git repo and CI

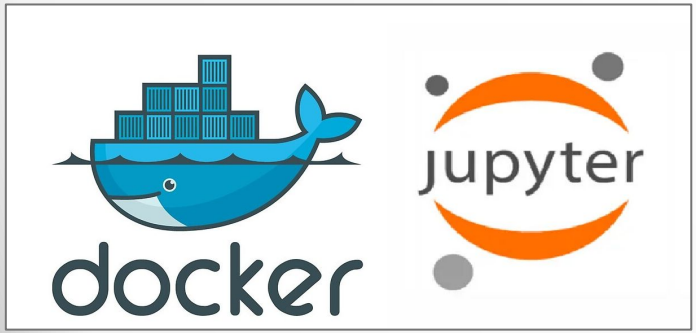




Rule 9: Connect Textual Statements to Underlying Results

Notebooks

- Notebooks provide a way to connect physics reasoning and code
- Can run notebooks in containerised analysis environments



Notebook example with MC

Lets look at some $B \to D\pi$ MC data using RDataFrame

- Plot the B and D mass

SWAN example

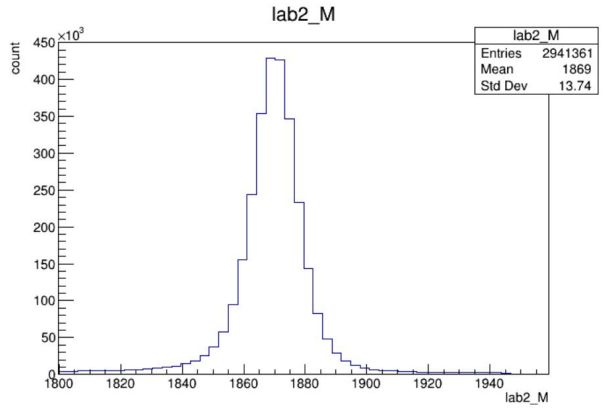
```
In [8]: import ROOT
ROOT.gErrorIgnoreLevel = ROOT.kInfo

files=["/eos/lhcb/wg/b2oc/TD_Bs2Dsh_Run2/MC_Run2/Stripping34_Sim09h/B2DX_MC_11264001_Bd_D-pi_2018_dw.root"]

ntupleName = "Bd2DPiOfflineTree/DecayTree"
fileName = files[:1]
dataframe = ROOT.RDataFrame(ntupleName,fileName)

canvas = ROOT.TCanvas()
Dmass=dataframe.Filter("lab0_M>4800 & lab0_M<6000").Histo1D("lab2_M")

Dmass.GetXaxis().SetRangeUser(1800, 2000)
Dmass.Draw()
canvas.Draw()
```



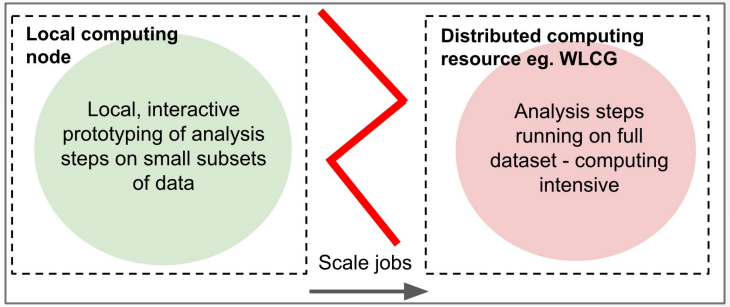
Rule 9: Connect Textual Statements to Underlying Results

Analysis facilities

"Infrastructure providing the data, software and computational resources to execute (an element of) an analysis workflow. … shared and supported through virtual organization."

154th LHCC Meeting

"The LHCC recommends that experiments engage in the process of developing and defining the structure of the future Analysis Facilities"

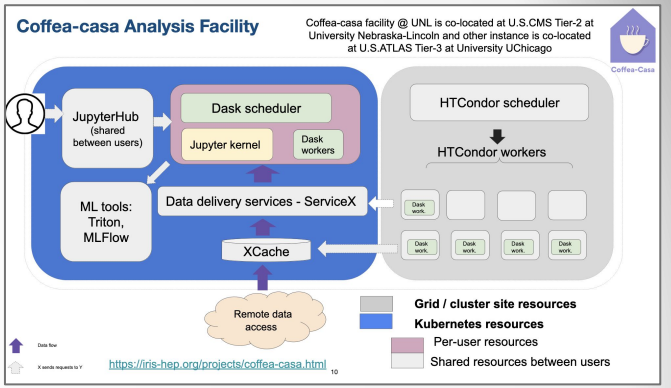


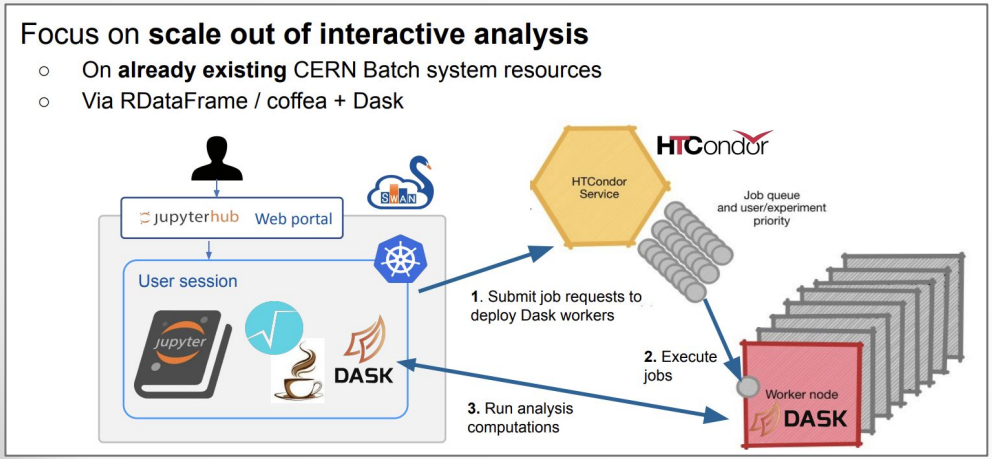
- AFs allow automatic, transparent **scaling** to batch resources from interactive session (notebook)
- Authentication, submission and retrieval abstracted away from user
- Results returned as if job was run locally

Rule 9: Connect Textual Statements to Underlying Results

AF infrastructure for ATLAS and CMS US analysts is advanced

- Jupyterhub (interactive notebooks)
- Integrated Dask scheduler for scaling to batch resources
- Token based AAI





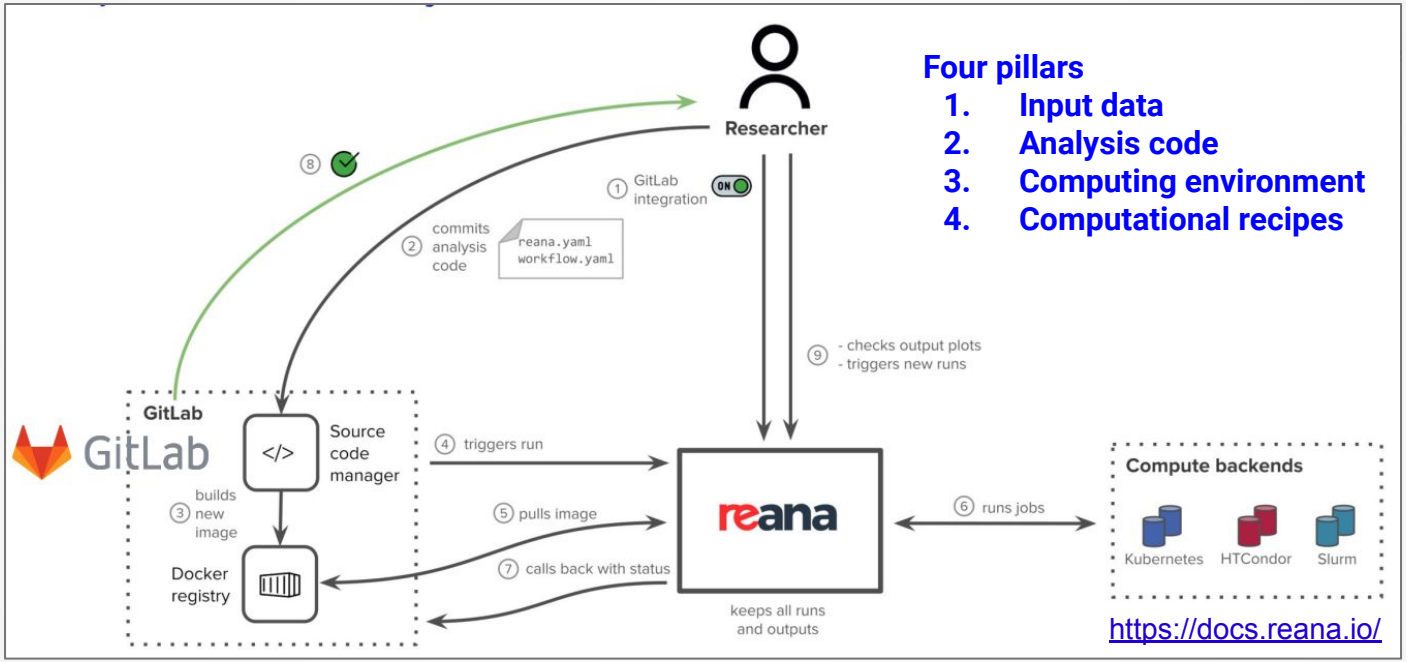
CERN's SWAN service has deployed **AF pilot**

"Interactive computing on big datasets, with analysis built on frameworks like RDataFrame and coffea"

SWAN AF talk at CHEP

Rule 10: Provide Public Access to Scripts, Runs, and Results

REANA - REproducible ANAlysis platform that can run containerised analysis pipelines on remote computing resources



AI to the rescue?

AI to the rescue?

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

def sawtooth_wave(x):
    return x / 3.14

x=[]
for i in range(100):
    x.append(-3.14+i/1000.)
x=np.array(x)
plt.plot(x, sawtooth_wave(x))
plt.show()
```

✓ 0.5s

AI to the rescue?

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

def sawtooth_wave(x):
    return x / 3.14

x=[]
for i in range(100):
    x.append(-3.14+i/1000.)
x=np.array(x)
plt.plot(x, sawtooth_wave(x))
plt.show()
```

✓ 0.5s

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad

def sawtooth_wave(x):
    """
    Generates a sawtooth wave function.

    Args:
        x: The input x-values.

    Returns:
        The sawtooth wave function.
    """
    return x / np.pi

# Define x-axis values
x = np.linspace(-np.pi, np.pi, 1000)

# Plot the sawtooth wave
plt.plot(x, sawtooth_wave(x))
plt.xlabel("x")
plt.ylabel("y")
plt.title("Sawtooth Wave")
plt.grid(True)
plt.show()
```

pi not hardcoded

Doc-string with purpose and arguments

Use of numpy arrays allowing vectorisation

Plot labels and formatting

AI to the rescue?

But not with the physics

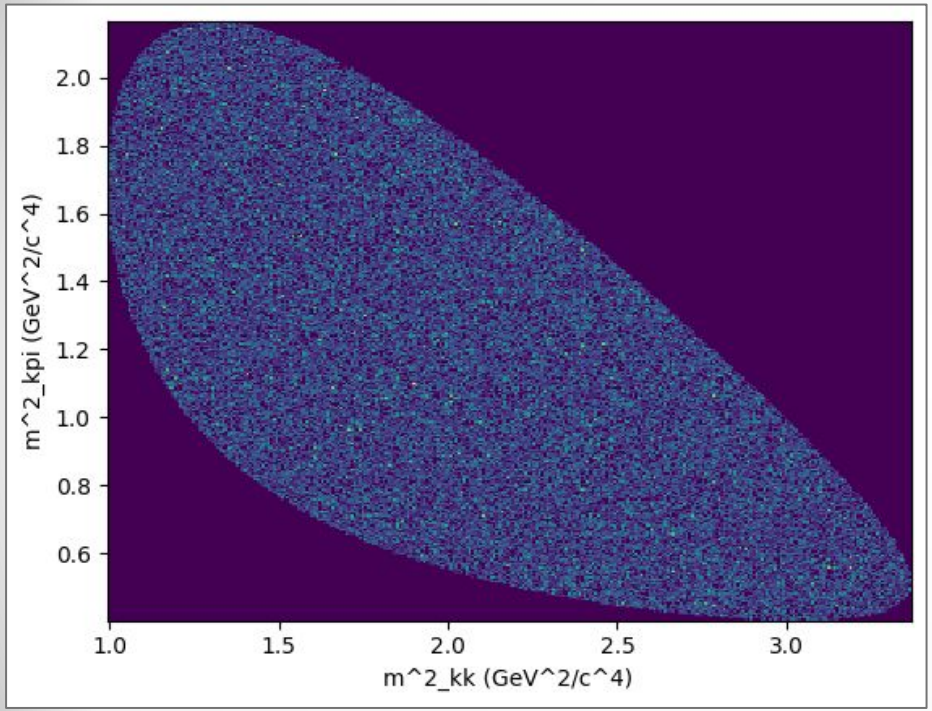
Task 1

Write a function(s) that generates n decay events of $D_s^+ \to K^+K^-\pi^+$ that are uniformly distributed over the allowed phasespace. The parameter n should be an argument of your function. Plot these events on a Dalitz plot for $n = 100000$ using a `hist2d` with 300x300 bins.

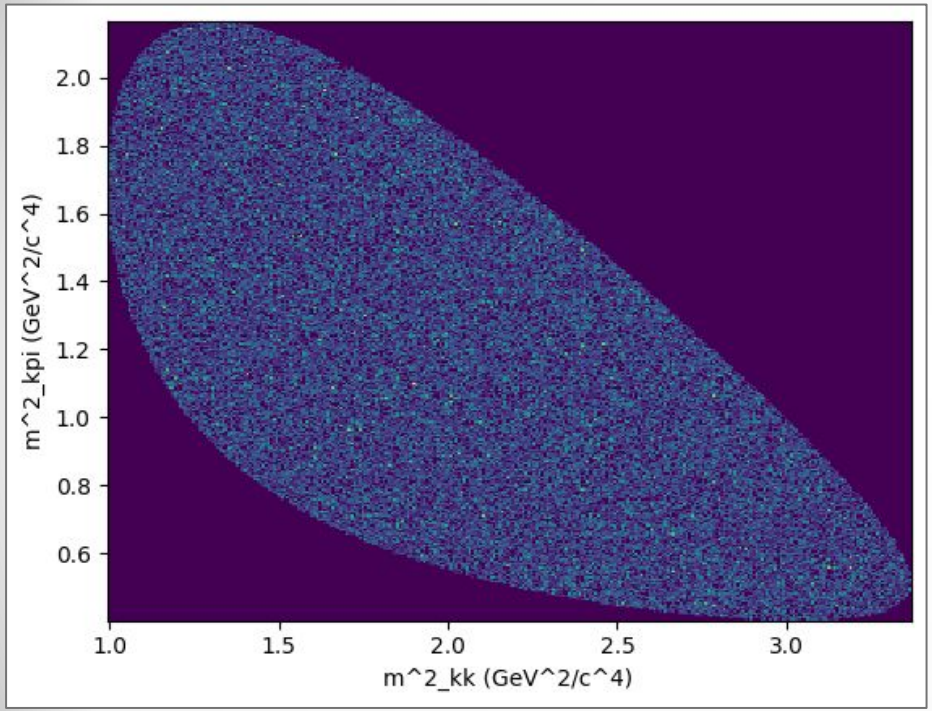
- "Allowed phasespace" here refers to the kinematically allowed region of the Dalitz plot, ie. the area shown in the example Dalitz plots above.
- Use the same units (GeV^2/c^4) and axes as in the example Dalitz plot on the left. To ease communication and make the code clearer we will use $d \to abc$ notation for the decay such that you will be working with the variables:
  - `m2ab` - the invariant mass squared of the K^+K^- system
  - `m2bc` - the invariant mass squared of the $K^-\pi^+$ system
  - `md, ma, mb, mc` - the masses of the decaying D_s meson and the 3 decay products respectively
- The masses you should use are
  - `m_dmeson=1.97`
  - `m_kaon=0.498`
  - `m_pion=0.135`

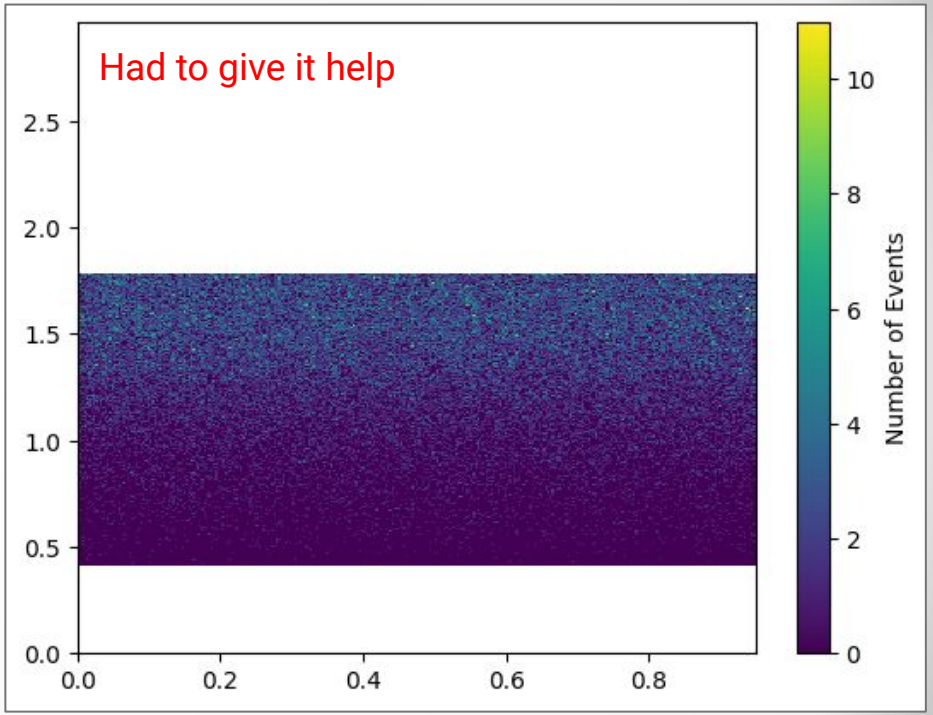
Hint: Use the accept-reject Monte Carlo method with NumPy functions. You only need to generate `m2ab` and `m2bc` values for the events. The range of `m2bc` values kinematically allowed given a `m2ab` value can be found in eqn 49.23a and 49.23b of the PDG review (note that the PDG uses $M^+ \to 1^+2^+3^-$ notation)

AI to the rescue?



AI to the rescue?



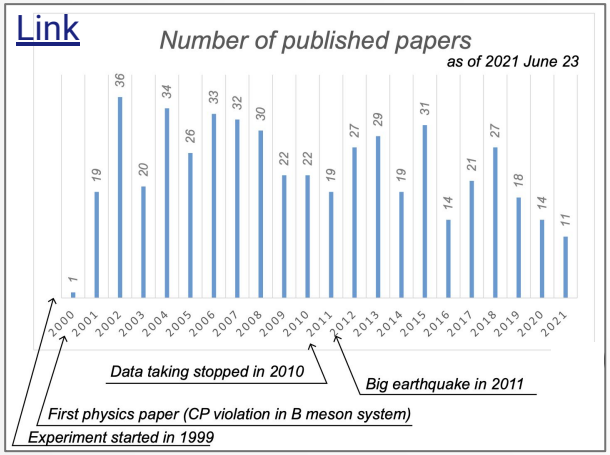


The long term…

The datasets we use are too valuable to not enable their full exploitation by **all**

- Some are completely unique and may remain so for some time eg. SMOG2
- Internal work on **preserving and opening** these must happen now

Belle I papers



Last month…

Search for $h_b(2P) \to \gamma\chi_{bJ}(1P)$ at $\sqrt{s} = 10.860$ GeV

Belle Collaboration • A. Boschetti Show All(142)

Oct 21, 2024

e-Print: 2410.16181 [hep-ex]

Report number: Belle Preprint 2024-07; KEK Preprint 2024-19

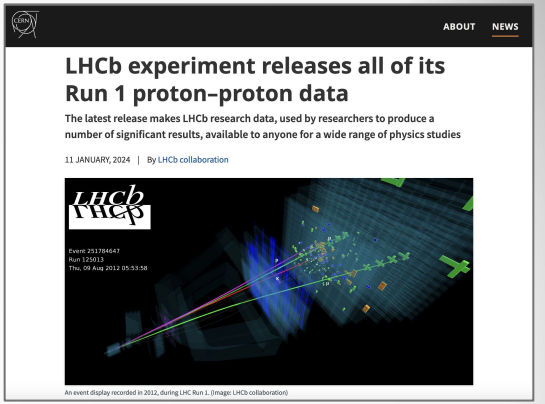
Experiments: KEK-BF-BELLE

View in: ADS Abstract Service

Making data open

The [CERN Open Data Policy] encourages the release of reconstructed data

- Data from all LHC experiments released through the Open Data Portal
- LHCb released its full Run 1 dataset ~ 800TB



Link

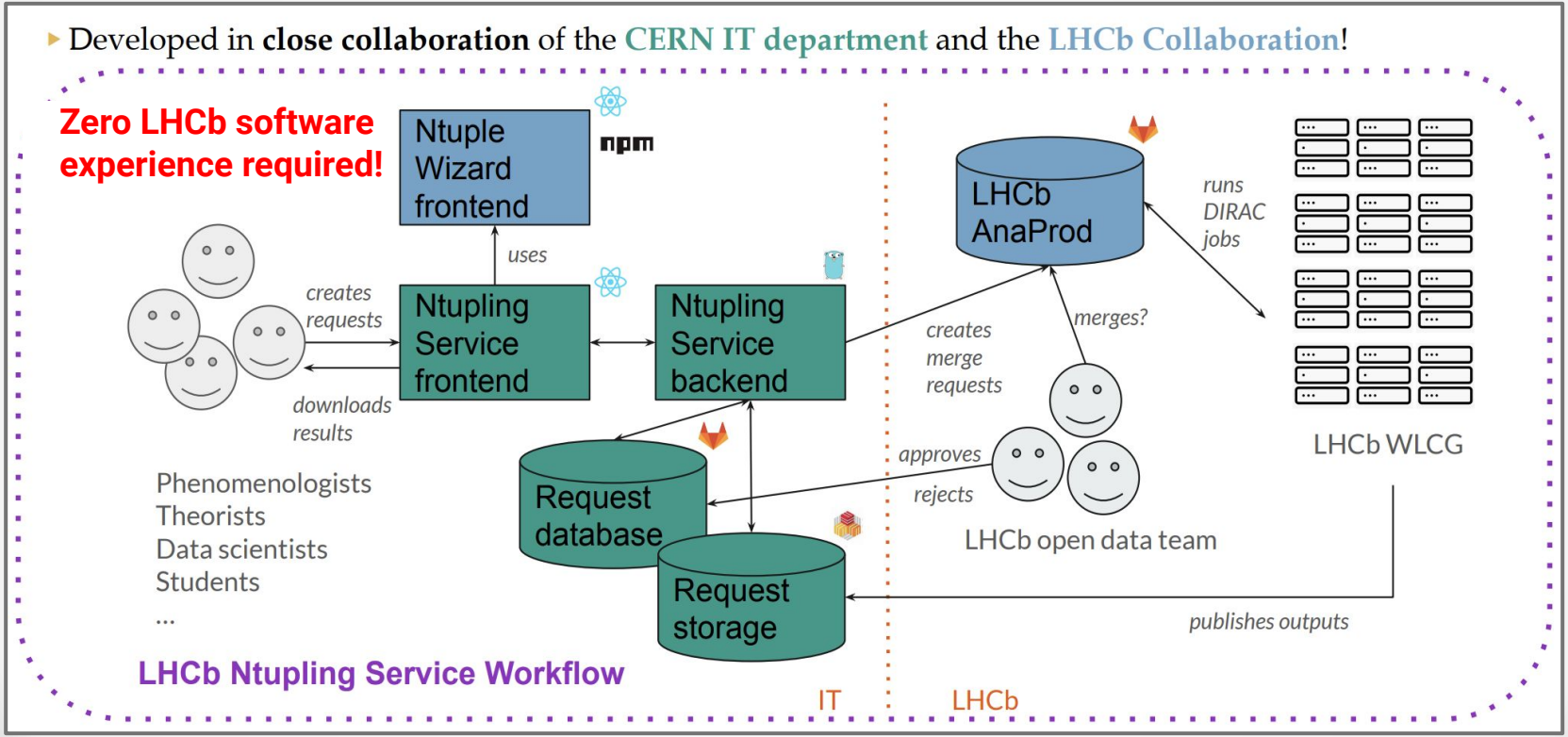
Releases for Run 2 and beyond impossible due to data volume - not scalable!

	ALICE	ATLAS	CMS	LHCb
Run-2	2 PB	0.5 PB	2 PB	10 PB (including Run-1)
Run-3	4 PB	1 PB	4 PB	45 PB
Total	**6 PB**	**1.5 PB**	**6 PB**	**55 PB**

Making data open

Scalable solution - **NTuple Wizard!**

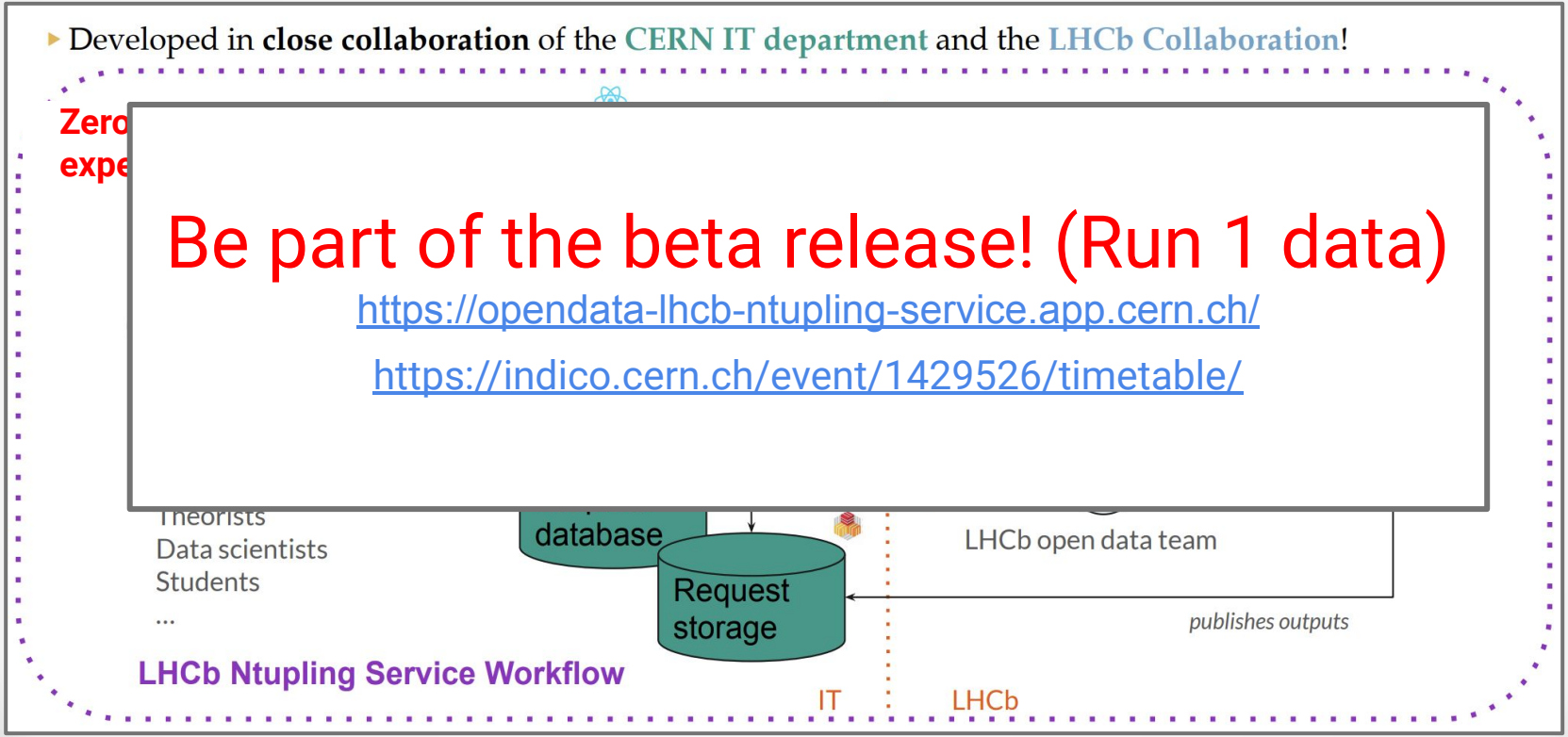
CHEP talk on NTuple Wizard



Making data open

Scalable solution - **NTuple Wizard!**

CHEP talk on NTuple Wizard



Tutorials

hsf-training.org/training-center/

- Git
- CI/CD
- Docker
- Singularity/Apptainer
- Reana
- Analysis essentials incl. Snakemake
- Julia

Self-study and in-person events

Close to home…

“Your closest collaborator is you six months ago…

but you don’t reply to email.”

Karl Broman

“Tools for Reproducible Research”

Backup

Analysis productions - declarative ntupling

Pre - run 3 analysts made their own ntuples

THE PROBLEM

- Submitting, monitoring and error handling O(10,000) grid jobs
- No data provenance
- Thousands of failing grid jobs

⇒ BIG barrier between analysts and data

Analysis productions - declarative ntupling

THE SOLUTION ⇒ Analysis productions

- Centralise and automate ntuple creation
  ⇒ Saves countless analyst-hours
- Exploit DIRAC transformation system
  ⇒ Full data provenance
- Full job testing on GitLab CI
  ⇒ No buggy jobs on grid

Simple yaml job configuration

```yaml
defaults:
  application: DaVinci/v64r10@x86_64_v2-el9-clang16-opt
  output: DATA.ROOT
  options:
    entrypoint: bs2dspi_run3.dv_simple:alg_config
    extra_options:
      input_raw_format: 0.5
      input_type: ROOT
      simulation: False
      data_type: "Upgrade"
      geometry_version: run3/trunk
      conditions_version: master
      input_process: "TurboPass"
      input_stream: "b2oc"
  inform:
    -
  wg: B2OC

{%- set datasets = [
 ('2024Data', 'MagDown', '24c2'),
 ('2024Data', 'MagUp', '24c2'),
```

What application to run

Job options

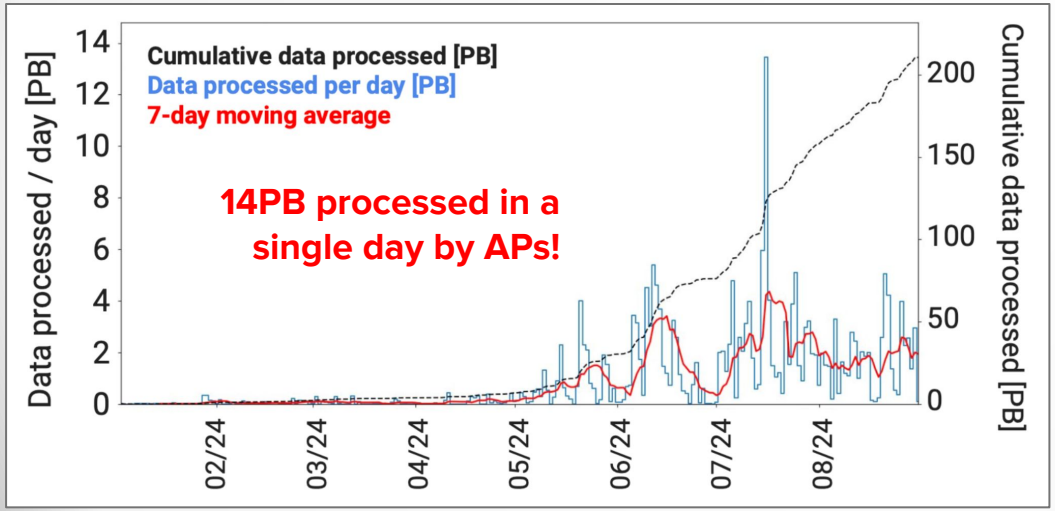
Data to run on

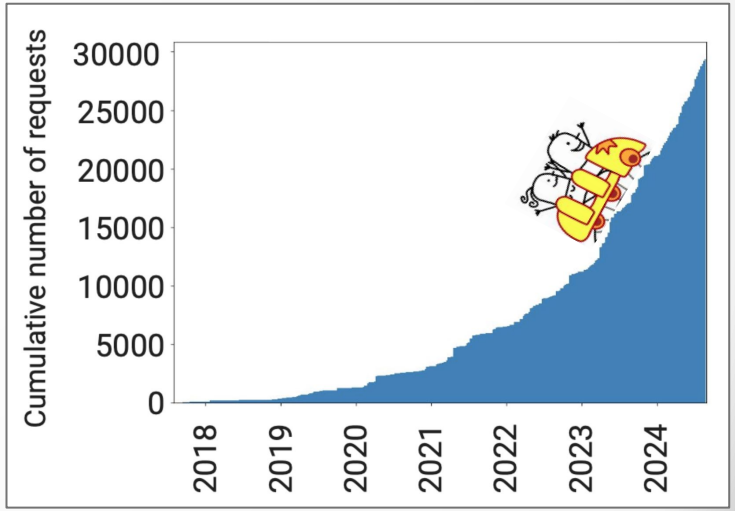
Use Jinja templating to “render” the YAML

Analysis productions - declarative ntupling

Full adoption of analysis productions at LHCb

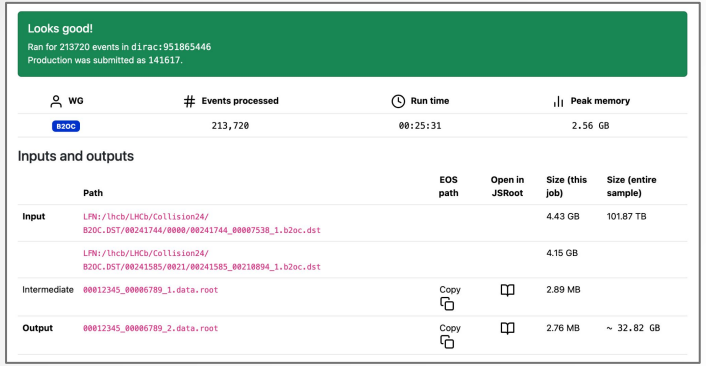
- Over 1200 Run 3 APs have been submitted so far
- 700+ “live” APs picking up data as it was Spruced
  - Analysts have been looking at data tuples days after it was recorded by detector
- We are making amazing use of the WLCG!



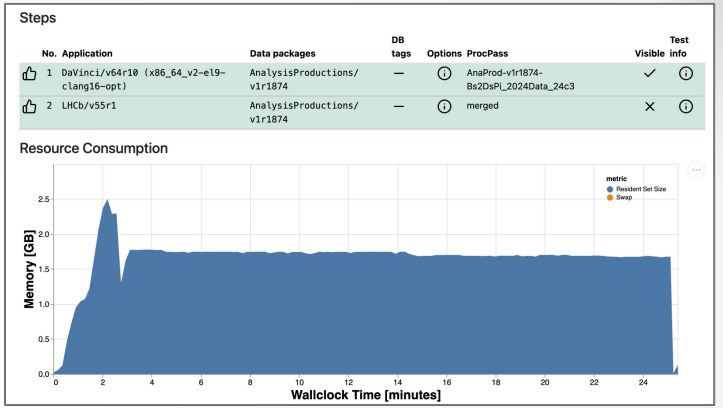


Analysis productions - declarative ntupling

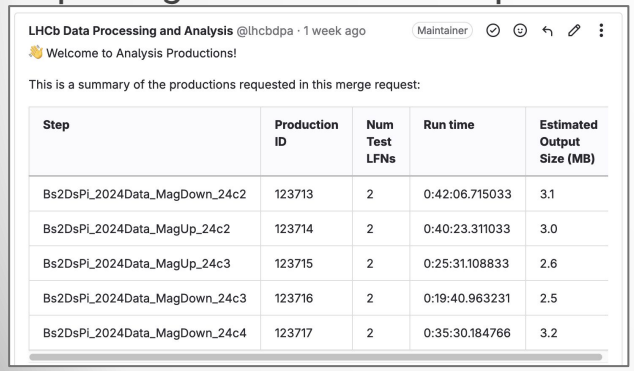
Comprehensive job testing through GitLab pipelines



Reporting on memory usage



Reporting on estimated output size



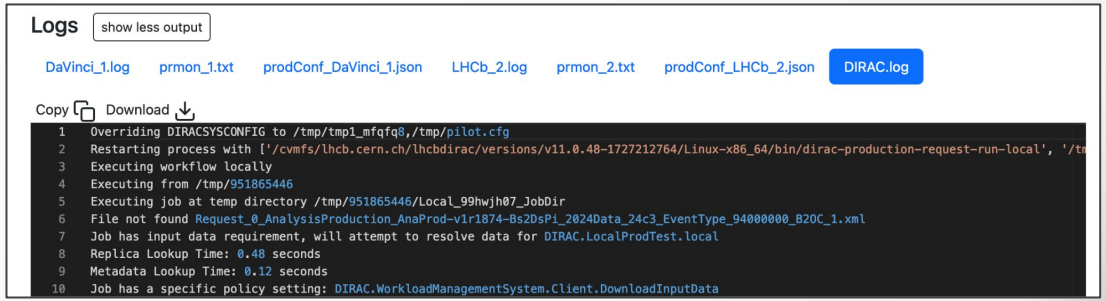
LHCb Data Processing and Analysis @lhcbdpa · 1 week ago Maintainer

👋 Welcome to Analysis Productions!

This is a summary of the productions requested in this merge request:

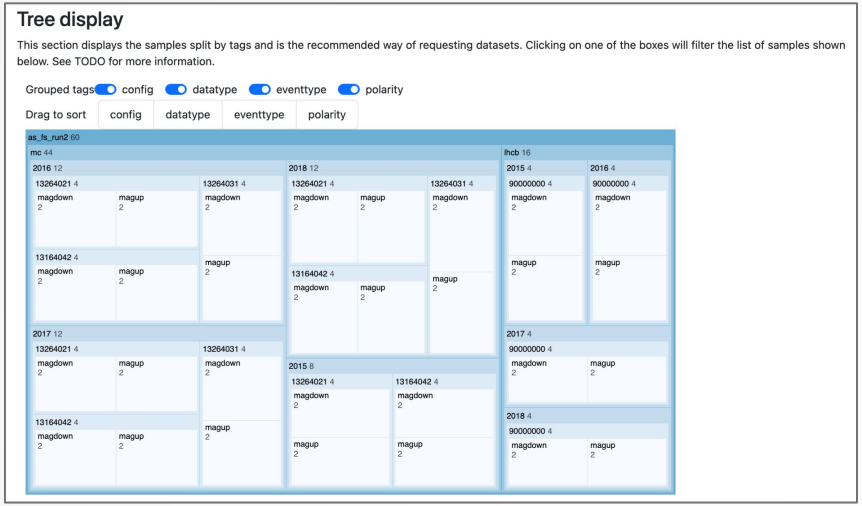
Step	Production ID	Num Test LFNs	Run time	Estimated Output Size (MB)
Bs2DsPi_2024Data_MagDown_24c2	123713	2	0:42:06.715033	3.1
Bs2DsPi_2024Data_MagUp_24c2	123714	2	0:40:23.311033	3.0
Bs2DsPi_2024Data_MagUp_24c3	123715	2	0:25:31.108833	2.6
Bs2DsPi_2024Data_MagDown_24c3	123716	2	0:19:40.963231	2.5
Bs2DsPi_2024Data_MagDown_24c4	123717	2	0:35:30.184766	3.2

Interactive logs with warning/error highlighting



Analysis productions - declarative ntupling

Full data provenance with datasets tagged by analysis



`apd` python packages allows for easy data file retrieval. Snakemake integrations!

```python
from apd import AnalysisData

datasets = AnalysisData("b2oc", "bs2dspi_run3")
bs2dspi_2024data_magdown_24c2_pfns = datasets(polarity="magdown", eventtype="94000000", datatype="2024")
```