PyHEP.dev 2024
Trip report

Vincenzo Eduardo Padulano, Juraj Smiesko
for EP-SFT

EP-SFT meeting 09.09.2024

The workshop

- 5-day workshop in Aachen, Germany
- Co-sponsored by ErUM-Data-Hub and IRIS-HEP
- Focused discussion sessions among HEP software developers
  - Every day different themes
  - Mostly Python oriented
- Around 25 people
- Daily program:
  - Self introductions
  - 1 or 2 presentations
  - Time for discussion and hacking

Monday - What is a HEP analysis?

- We defined the scope of the discussions: what “HEP data analysis” means
  - “**transforms** raw measurements into **human-interpretable formats**, such as tables and visualizations, towards the extraction of physics quantities of interest. The process includes summary statistics, statistical inference, and machine learning. The act of analyzing data is highly **iterative**, changing strategy in response to partial results.”
- Analysis Grand Challenge re-focused on a new activity called “200 Gbps Challenge”
  - Second AGC analysis benchmark postponed until after the new activity ends
  - Architecture of the computing site heavily influences its performance
  - AGC may provide blueprints for computing sites
- A demo of the AGC using dask-awkward + coffea 2024 was shown
  - A very large Dask computation graph is produced, which introduces memory issues

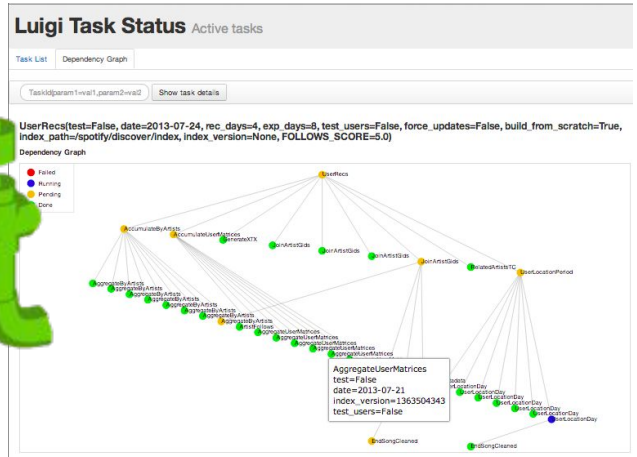
Tuesday - likelihood building, estimation, serialisation

- Existing packages, open-world vs closed-world
  - Some packages are the basis for many others: RooFit, pyhf, zfit
- Minimization is crucial for point estimates, but there are different stopping criteria depending on the minimizer
  - Minuit2 is the most widely used
- Would be beneficial to design a common interface for likelihood estimation (a-la UHI)
  - Standardise call signature, parameter configuration and return information
  - Would allow running same inference workflow with different tools
- The HS3 common serialisation format is highly praised
  - It was also discussed an hypothetical computation serialisation format (a-la LLVM IR)
  - Easier preservation and sharing among different groups/experiments

Wednesday - workflow management and histogramming

Workflow management system: software to describe, manage and execute arbitrary workloads

- Most common libraries: Luigi, SnakeMake, AirFlow, Dask
- Luigi-based tools: b2luigi (used in Belle II) and LAW
- Useful tools for reproducibility and maintainability of analysis workflows



Wednesday - workflow management and histogramming

Histogramming:

- Sckit-HEP-based packages use the Python bindings of boost histogram
- UHI has enabled different higher-level histogram production tools to coexist
- Discussion on histogram filling focused mostly on GPU implementations, whereas CPU filling “looks solved”.
- But storage can still be challenging
  - Rectangular n-dim array of values vs sparse arrays
  - Store via formats which allow chunking, compression and growable data structures (e.g. HDF5, zarr, blosc2)
- Accumulation only becomes challenging with very large histograms
  - Can leverage parallel and distributed accumulation strategies
  - Also direct accumulation to disk if data format allows
- Plotting with ROOT graphics is well established, but currently incompatible with the matplotlib-based graphical backends of scikit-HEP packages

Thursday - tools for analysis (at scale)

- FCCAnalyses/Key4hep was presented here
- Julia interoperability with Python
  - Usage through “string” evaluation
  - Julia is missing package manager able to integrate external libraries
- PocketCoffea framework continues to be developed
  - Framework to configure processing of CMS nanoAOD datasets
- Functional-programming analysis language in Python: Lena
  - One defines their analysis as a sequence of loosely coupled analysis elements

Friday - wrap up

- Friday morning spent writing the workshop report - all participants together
  - To be published soon on arXiv
- Discussed PyHEP.dev 2025 plans
  - Most probably during the same week and in the same location of SciPy 2025 (Tacoma, WA)
  - Workshops will continue to rotate between the United States and Europe each year
  - The PyHEP.dev organizers want to further interactions with the wider SciPy community

Conclusions

- The PyHEP community is actively working onwith standards for interoperability among different parts of HEP analysis
  - E.g. HS3, UHI
  - Support in ROOT is either already available (HS3) or planned (UHI)
- This type of effort has the potential outcome of allowing to mix and match different tools according to analysts' preferences
  - And also different programming languages (C++, Python, Julia…)
- The format allowed for focused and fruitful discussion with community experts