Python in HEP
first workshop, community activities

Eduardo Rodrigues
University of Cincinnati

LHCb Tuesday Meeting, CERN, 19th Sep. 2018

Popularity of Programming Language
http://pypl.github.io/PYPL.html
Python, you say?

- Taken from opening talk at PyHEP 2018 workshop, 7-8 July, Sofia, Bulgaria
  

- Starts to illustrate why I organised a first workshop to look at the role of Python in HEP … See later …
How’s the Python scientific ecosystem like, outside HEP?

See the Scikit-HEP project [GitHub](https://github.com/scikit-hep) for a HEP domain-specific community effort …

---

Jake VanderPlas, *The Unexpected Effectiveness of Python in Science*, PyCon 2017
The Python scientific stack really is important, even more if you are thinking about a career outside HEP ...

And since we talk a lot on how to help young people, training on (at least some of) these tools should be seen as very relevant.

Taken from: figure eight, Data Scientist Report 2018 (full report)
Outside HEP

- Has become the lingua franca for data science and machine learning
- Traditionally, emphasis on developer productivity over code runtime
- Steering high performance backends gives excellent performance for the right problems

HEP

- Python is a first-class language in HEP
- Very popular in analysis and job configuration
- HEP exploits this route to some extent, but much more to/can be done

See also discussions on need for adequate Python support in the HSF Community White Paper (CWP)


and the supporting paper from the CWP Analysis & Interpretation WG,
Remember:
Hans Dembinski conducted recently an online ROOT survey among LHCb analysts
• Use of ROOT software
• Primary sources for learning and debugging
• Positive and negative experiences with ROOT
• Free comments

⇒ 74 responses received

 ROOT from Python is just as used as is plain C++!

Taken from
Hans Dembinski, User Feedback from LHCb, ROOT Users’ workshop, Sarajevo, Sep. 2018
Workshop raison d’être and goals, in brief

- Step back and review evolution of Python in the HEP community at large
  - There are certainly HEP conferences & workshops discussing computing & software
    but none really devoted to this critical language in analysis

- Python clearly identified as first-class language during the Community White Paper process

- Need to consolidate this consensus and plan the future directions
  - Where we are going, want to go, need to improve
  - Tools usage, needs and developments, training and education, which Python, etc.

- Bring together users and developers from a wide audience

- Educative, not just informative, workshop,
  with lively discussions in the many free and dedicated time slots we foresaw
1.5 day workshop
Pre-CHEP2018 event @ Sofia, Bulgaria

Indico agenda – for all details
70 participants

We had a very diverse set of participants
- BTW, excellent contingent from LHCb 🙌!

(Taken from the pre-workshop questionnaire)
Organisation not totally standard – worked really well

1. Pre-workshop questionnaire
   - To understand the background, interests and concerns of those coming to the workshop
   - We hoped it would guide the topics we addressed, and we think it has validated what we put on the agenda
   - And could stimulate some discussion…

2. Workshop
   - Presentation and discussion of results of pre-workshop questionnaire
     at 1st session “Historical perspective / overview”
   - 7 Sessions, all plenary,
     including an open discussion on education and training
   - Live notes taken during the sessions,
     which provided plenty of food for thought

3. Post-workshop survey

Workshop topics / sessions:
- Historical perspective / overview
- HEP python software ecosystem
- Analysis & HEP frameworks
- PyROOT and Python bindings
- Distribution and installation
- Python 2 to 3
- Open discussion on education and training
- Keynote presentation on JupyterLab
- 71% of workshop participants responded

- See details in

---

**Motivation to use Python**

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Important to</th>
<th>%age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development speed and efficiency</td>
<td>40</td>
<td>80%</td>
</tr>
<tr>
<td>Availability of other software packages</td>
<td>26</td>
<td>52%</td>
</tr>
<tr>
<td>Interface language</td>
<td>20</td>
<td>40%</td>
</tr>
<tr>
<td>Machine learning packages</td>
<td>19</td>
<td>38%</td>
</tr>
<tr>
<td>I just like it</td>
<td>12</td>
<td>24%</td>
</tr>
<tr>
<td>They make me do it...</td>
<td>5</td>
<td>10%</td>
</tr>
</tbody>
</table>
What are we using Python for?

How would you characterise your principle use of Python?

- 32.7%: Lots of physics (no surprise)
- 18.4%: ML, Experiment Production, Physics Analysis
- 16.3%: More infrastructure use than we expected
- 20.4%: But plays to Python’s strengths, of course
2, 3, 2.5?

- Amongst us there is a very healthy use of Python 3
- Both 2 and 3 we interpret as “3 when I can, 2 if I have to”
- Migration to Python 3 is a big concern for the community as we’ll see later
Use and Evolution

What fraction of your programming is done in Python?
50 responses

- <33%: 36%
- 33% to 66%: 42%
- >66%: 22%

• We do use Python a lot (selection bias!)

In the future, how would you like your use of Python to evolve (as a fraction of your programming time)?
50 responses

• Anticipating that will stay the same or increase
## HEP Investments for the future?

<table>
<thead>
<tr>
<th>Important Development / Investment for HEP</th>
<th>Important to</th>
<th>%age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration from Python 2 to 3</td>
<td>29</td>
<td>58%</td>
</tr>
<tr>
<td>Better ROOT integration</td>
<td>18</td>
<td>36%</td>
</tr>
<tr>
<td>Better experiment software integration</td>
<td>14</td>
<td>28%</td>
</tr>
<tr>
<td>Better development tooling</td>
<td>13</td>
<td>26%</td>
</tr>
<tr>
<td>Improved training</td>
<td>12</td>
<td>24%</td>
</tr>
<tr>
<td>Runtime speed improvements</td>
<td>9</td>
<td>18%</td>
</tr>
</tbody>
</table>
Training Needs

For training, what do you think is the most urgent training to develop and give in HEP?

48 réponses

- Very strong interest in Python data analysis ecosystem
- Open question is what is the role of HEP in this sort of training?
- In particular what’s the boundary between HEP specific topics and generic ones?
PyHEP 2018 Workshop – pre-workshop questionnaire

- Significant use of Python for a significant fraction of analyses

- Surprisingly little correlation with experiment (caveat for this one question: small statistics)

If you do physics analysis, to what extent (%) do you use Python, instead of compiled/interpreted C++, in your analysis work? 32 Responses

![Histogram showing the percentage of analysis done in Python]

![Box plots showing the percentage of analysis done in Python for different experiments]

Eduardo Rodrigues

LHCb Tuesday Meeting, CERN, 18th Sep. 2018
Overview

- Python is on an upward trajectory
  - Data science, Machine learning providing strong drivers
- HEP usage is increasing too
  - Coupled to expansion of Python ecosystem, but also building on Python’s traditional strengths
- Notebooks are a huge hit
  - Many thanks to Vidar for the JupyterLab talk
- Pre-workshop questionnaire
  - Training
  - Plotting
    - Galleries are really useful to find examples, didactic too!
  - Installation
Inventory

- Even we did not know what useful packages are available
- Inventory of tools appropriate to HEP would be great
  - With notebooks and galleries to show how to use them
- Orphaned packages, but still useful?
  - Way to look for a maintainer
  - Scikit-HEP has handed over packages between maintainers
- Repository of expertise in the PyHEP community
  - Ties well with hot topic of education and training across the field
Experiment and Analysis Directions

- Python is a language that can be used for all the computing
  - End to end
  - Flexible
  - Naturally modular
- The Belle II analysis/training jupyter cluster looks great
- Extensions to full analysis clusters?
  - Need good integration with storage
  - SWAN as an integrated, stable and reproducible environment

*This looks like a key direction, aligned with Community White Paper Roadmap*
PyHEP 2018 Workshop – summary and outcomes

ROOT

- Data model ideal for HEP
- Fitting, histograms best in class
- Heavy component
  - Too burdensome for some small experiments it seems
  - Modularity would help
- Easier ways to install
  - NLeSC effort was greatly appreciated
- cppyy is a contribution that is far less well known than it should be
- PyROOT developments exciting
  - Particularly adding pythonisation, to make things natural

Our vision:

Python
ROOT

Python
ROOT

Python
ROOT

Time
Training

- The Belle II analysis jupyter cluster looks great
- Extensions to analysis clusters?
  - Need good integration with storage
  - Cf. SWAN
- Using standard Python libraries to achieve HEP workflows is a concern for our community
- Training session discussion was brilliant, right!
  - (What do you mean you forgot already what was said?)
Distribute and Install

- Can we be as standard as possible?
  - CMS using pip + PyPI
- Distributing whole HEP stack is a difficult problem
  - Worse than Python, multidimensional
- Distinguish experiment stack from analysis
  - Toolboxes, not frameworks
  - SWAN encapsulates things really well
  - Daring view: ubiquitous network access + browser...
- Modularity and flexibility of the solution vital
  - HSF Packaging Group should pay more attention to this
To Python 3

- Will be painful for the large pieces, but we just have to do this
  - LS2 project for LHC experiments - having to get to the end of Run 3 with an unsupported Python would be uncomfortable
  - An increasing gap between legacy Python 2 and Python 3 would hurt
PyHEP 2018 Workshop – post-workshop survey

- Standard survey to assess level of satisfaction of participants

**Main conclusions**

- Participants largely happy
- Interest in having more PyHEP workshops

- Mild interest in occasional 1-hour PyHEP meetings organized by the HSF
Format of future PyHEP workshops (participants picked 2 preferred options)

Only general overview presentations, albeit topical,
In 1-2-day workshops

Inclusion of keynote presentations on hot topics

Longer workshop with topical talks and related training sessions

Longer workshop with topical talks and a related hackathon
“Python in HEP” community
Building a community - communication

Community "start-up"

- Obvious interest from PyHEP 2018 workshop participants in building a community of developers and users
- Created a (non-formal) coordination team to push activities forward
  - Careful selection of team, to embrace Particle Physics at large
  - So far: Eduardo (LHCb & DIANA-HEP), Graeme Stewart (ATLAS & HSF), Jeff Templon (Nikhef & Grid computing), Chris Tunnell (XENON1T). Person from neutrino community to join soon.
- Fully in line with the HSF activities & interests, with HSF support

Actions, please!

- Need a low entry-level and informal way to communicate and exchange ideas, material, etc.
- We created a “Python in HEP” Gitter channel
  - Well over 100 messages exchanged already, on a diverse set of topics, from general to technical!
- Free and trivial sign-in … (e.g. with a GitHub or GitLab account)
Building a community – training & education

- Seen as a very important topic
- General interest in sharing material and knowledge, organise events across experiment boundaries

**Actions, please!**

- We created an HSF organisation on GitHub for training and education material: [https://github.com/hsf-training](https://github.com/hsf-training)

  ![HSF Training Material](image)

- 2 repositories exist as of today:
  - Good-old LHCb “analysis essentials” course moved to this HSF organisation,
    see [https://github.com/hsf-training/analysis-essentials](https://github.com/hsf-training/analysis-essentials),
    to push on a community-wide effort around training and education
  - New repository for “Python in HEP” resources: [https://github.com/hsf-training/PyHEP-resources](https://github.com/hsf-training/PyHEP-resources)
Software – recognition and citations

Recall of work in software & computing

- Recurrent topic, even more following the preparation and release of the HSF Community White Paper
- Still a long way to go for software work to be on the same recognition level as analysis and hardware work!

Actions, please!

- Please cite software-related publications and software packages you use!
- As far as the PyHEP 2018 workshop goes:
  - We created a PyHEP2018 community on Zenodo, to collect all talks (there are no proceedings)
  - Each uploaded presentation gets attributed a DOI, for standard citation - nice and easy
In short

- Feeling the urge to connect and contribute?
- We are a community ⇒ everybody welcome!
- Easy, just join and start contributing … much to take in and give out!

Links

- Gitter channel [HSF/PyHEP](#)
- GitHub repository “Python in HEP” resources

Mailing lists

- HSF general forum [hsf-forum@googlegroups.com](#)
- PyHEP coordination team: [hsf-pyhep-organisation@googlegroups.com](#)

Thank you