(Selected) results from HSF training survey

David Lange
March 22, 2019
Survey goals and timeline

• To get input on training needs and current training practices for various topics in HEP related to software/computing and related software-centric areas. This information will potentially be used to seek resources and to organize training activities to meet those training needs.
  • Discussed in HSF meetings starting in October
  • Opened/announced: January 4
  • Closed: February 1

• 344 people responded! Thanks to all who participated and who helped spread the survey to their communities
Who responded
Who responded

Mostly students and postdocs

Mostly LHC researchers
Who responded

Consistent with students and postdocs
Questions we asked

1. Knowledge and frequency of use of some scientific software/computing tools and areas of research [Select options from a list of areas]

2. In which areas do you see the need for more training materials/courses to help your research? [Select options from a list of areas]

3. Which training sources have you previous used for learning scientific software and methods, and how effective was each training format for you? [Answer for each option on list of sources]

4. In which periods would you be able to attend a remote training course?
   - Consensus answer: Anytime
Knowledge and frequency of use of some scientific software/computing tools

This was unfortunately asked as two questions in one

<table>
<thead>
<tr>
<th>Most used</th>
<th>Least used</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>TensorFlow/Theano/Keras</td>
</tr>
<tr>
<td>Python</td>
<td>SciKit-Learn</td>
</tr>
<tr>
<td>Git (GitHub, GitLab)</td>
<td>Programming for accelerators (GPUs, FPGAs)</td>
</tr>
<tr>
<td>C++</td>
<td>Geant4</td>
</tr>
<tr>
<td>Statistical methods</td>
<td>TMVA</td>
</tr>
<tr>
<td>RooFit</td>
<td>Big Data Tools (Spark, etc.)</td>
</tr>
<tr>
<td>Random Number Generators</td>
<td>Other languages (Fortran, Scala, etc.)</td>
</tr>
<tr>
<td>Notebook applications (Jupyter, etc)</td>
<td>java</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>julia</td>
</tr>
<tr>
<td>Statistical toolkits</td>
<td></td>
</tr>
<tr>
<td>Pandas</td>
<td></td>
</tr>
<tr>
<td>Machine learning tools</td>
<td></td>
</tr>
<tr>
<td>Jenkins/Travis or other continuous integration systems</td>
<td></td>
</tr>
<tr>
<td>Machine Learning methods</td>
<td></td>
</tr>
<tr>
<td>Parallel Programming (Multicore/HPC)</td>
<td></td>
</tr>
<tr>
<td>Software Performance Measurement/Optimization</td>
<td></td>
</tr>
<tr>
<td>Event generators (Pythia8, Madgraph, Sherpa, etc)</td>
<td></td>
</tr>
</tbody>
</table>

Fraction of answers

Don't use 1/year 1/month 1/week 1/day
Areas that need for more training materials/courses to help your research
Any correlations between use/knowledge and need of materials?

- Some outliers with frequent use / good knowledge but low training need (beyond current status)
  - Random # generators
  - Notebooks
  - ROOT
  - Git
Training sources have you previous used for learning scientific software and methods

- Written material in textbooks, workbooks, webpages, Stack Exchange, etc.
- University Course on Programming (Fortran, C, C++, Java, Python, etc.)
- Video recorded lectures or presentations
- University Course on Computational Physics or Computational Methods
- Experiment organized software training (CMSDAS, LHCb Starterkit, etc.)
- University Computer Science courses (types other than the ones above)
- University mini-courses (e.g. 1/2 or 1 day topical courses)
- MOOCs or other self-paced online training
- Other HEP organized school / course
- University Course on Software Engineering and/or Performance Optimization
- Software or Data Carpentry workshop
- Hands-on sessions at DS@HEP, IML or other HEP workshops/conferences
- CERN summer student programs
- Industry (Intel, NVIDIA, etc.) training
- CERN school of computing
- FNAL LPC HATS Training
- MLHEP School
- CoDas-HEP school
- INFN ESC School (Bertinoro)
- GridKa School

Most used

Least used

Not effective
Effective
Very effective
Not used
Comments received: Some areas of interest that we left out of the survey (directly or indirectly)

1. Debugging tools
2. Go
3. Object oriented programming, encapsulation
4. Using containers
5. Labview
6. SCRUM and Agile methods to organize software dev collaboration
7. Software engineering methods, software project management and the software development process
8. Data structures
9. DAQ software
10. Shell scripting
11. Basic functionality of CPU, RAM, disks, network
Comments about training courses

• Make them more interactive and fun.
• More starter kit format,
• Start from "Hello World!" and build from there. Not "Hello World!" and now you're ready to run TensorFlow on your own.
• Some intro courses are too advanced for beginners
• Instead of abstract learning of topics and concepts, more applicative workshops will be better.
• More data science tools
More comments

• Forum for weekly training / discussions
• Online forum
• Training opportunities are important but are still quite limited
• I think there are no needs. If a person is interested, he will be able to find a lot of helpful information himself. If one is not interested, any courses are useless.
• Holding training sessions focused on programming (e.g. modern C++, software design, accelerator frameworks like SYCL, concurrency, library/API design). I found the courses and content at Cppcon to be really helpful in this regard.
Thank you