Software and Computing Training at Jefferson Lab

Ole Hansen

Jefferson Lab

HOW 2019
March 21, 2019
The Software & Computing Landscape at Jefferson Lab

- 4 halls → 3 different simulation & reconstruction packages

<table>
<thead>
<tr>
<th>Hall</th>
<th>Simulation</th>
<th>Reconstruction</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &amp; C</td>
<td>SIMC (Fortran)</td>
<td>Podd (ROOT/C++)</td>
<td>ROOT/C++</td>
</tr>
<tr>
<td>B</td>
<td>GEMC (Geant4)</td>
<td>CLARA (Java/C++)</td>
<td>Analysis Studio (Java)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clas12Tool (ROOT/C++)</td>
</tr>
<tr>
<td>D</td>
<td>HDGeant4 (Geant4)</td>
<td>JANA/DANA (C++)</td>
<td>ROOT/C++</td>
</tr>
</tbody>
</table>

Computing

- Batch farm w/in-house workflow tools (SWIF)
- Some OSG, NERSC (experimental)
4 halls → 3 different simulation & reconstruction packages

<table>
<thead>
<tr>
<th>Hall</th>
<th>Simulation</th>
<th>Reconstruction</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &amp; C</td>
<td>SIMC (Fortran)</td>
<td>Podd (ROOT/C++)</td>
<td>ROOT/C++</td>
</tr>
<tr>
<td>B</td>
<td>GEMC (Geant4)</td>
<td>CLARA (Java/C++)</td>
<td>Analysis Studio (Java)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Clas12Tool (ROOT/C++)</td>
</tr>
<tr>
<td>D</td>
<td>HDGeant4 (Geant4)</td>
<td>JANA/DANA (C++)</td>
<td>ROOT/C++</td>
</tr>
</tbody>
</table>

Computing

- Batch farm w/in-house workflow tools (SWIF)
- Some OSG, NERSC (experimental)

What to teach? C++, ROOT, maybe PyROOT, SWIF
JLab Training: General Beginner/Intermediate

- Basic skills: Software Carpentry Workshops 2017/2018
  - For students/beginners
  - Unix shell, Git, Python, ROOT, Jupyter notebooks, OSG/HTCondor

- Hall A/C
  - 2-day software workshops, for beginning graduate students, new postdocs
  - Hands-on tutorials in Virtual Machine
  - Hall A/C simulation & reconstruction software, analysis methods
  - Generally useful: ROOT, Python/PyROOT, Git, JLab batch farm use/SWIF
Hall A/C Workshop 2018

Joint Hall A & C Data Analysis Workshop June 2018

June 25-26, 2018, ABC Auditorium, Jefferson Lab

Resources
- Software Setup Instructions
- Virtual Machine Image download
- GitHub repository of tutorial support files
- List of participants

Program
Monday, June 25, 2018

Morning Session
(Chair: Mark Jones)
- Video Recording

General
09:00 — Welcome — Ole Hansen
09:05 — Overview & Update on Hall A Analysis Software — Ole Hansen
09:30 — Overview & Update of the Hall C Analyzer — Eric Pessaer

Hall A Analysis
10:00 — Hall A python optimization — Tsung Fu
10:30 — Optics for intensified spectrometers — Eric Christy
10:45 Coffee Break
11:00 — Beam energy determination — Doug Hoppesham
11:30 — Using ROOT, databases in analysis (Part 1) — Shinya U
12:00 — Software reply on farms, analysis organization — Tyler Rogers
12:30 Lunch (on your own)

Afternoon Session
(Chair: Ole Hansen)
- Video Recording

Software and Computing Training at Jefferson Lab

HOW 2019, March 21, 2019 4 / 9
Hall-Specific Intermediate/Advanced

- **Hall B**
  - 1–3 hour _hands-on tutorials in Docker containers_ at collaboration meetings
  - Very specific to Hall B environment
  - Most recent: [https://www.jlab.org/indico/event/303/](https://www.jlab.org/indico/event/303/)

- **Hall D**
  - 2-day Analysis Workshop 2013:
  - 3-day Physics Workshop 2016:
  - Hands-on tutorials in Virtual Machine
  - Limited to Hall D collaboration (contents and access)
General Intermediate/Advanced/Expert

- Geant4
  - 5-day hands-on course by Geant4 developers
  - Offered annually at changing locations
  - 2012 workshop at JLab: https://www.jlab.org/conferences/geant4/

- Computing Roundtable
  - Advanced presentations on recent developments
  - Machine learning, new programming tools, languages, libraries
  - 2018 series: https://www.jlab.org/indico/event/247/

- Machine Learning
  - Informal weekly lunch meetings for anyone interested
  - Tracking ML meeting series (https://github.com/JeffersonLab/trackingML)
Observations

- **Useful Teaching Tools**
  - Fully configured environments (VMs, containers)
  - Hands-on exercises/tutorials combined with concepts overview
  - Alerts about common pitfalls
  - AV recordings of presentations

- Teaching of **analysis techniques** as important as technical training

- Huge variation in student/postdoc preparedness

- 2–5 day workshops or “schools” are invariably too short

- Preparing workshops can be enormously time-consuming

- **Good written documentation** (*not* the auto-generated kind) often best for in-depth learning
Not Covered—A Very Incomplete List

- **Technical**
  - C++ (incl. new standards, STL)
  - Intermediate/advanced ROOT
  - In-depth Python, libraries (numpy ...)

- **Logistical**
  - Coordinated effort across the lab
  - Modern teaching tools (e.g. more Jupyter Notebooks, web-based courses)
Plans

- 5-day workshop, graduated by skill level, maybe 2020, hopefully across halls