EPJ Web of Conferences **295**, 08007 (2024) https://doi.org/10.1051/epjconf/202429508007
CHEP 2023

DUNE Computing Tutorials

David DeMuth, Jr.[1,*], *Heidi* Schellman[2], and *Clare* David[3],
on behalf of the DUNE Collaboration

[1]Valley City State University, 101 College St. SE, Valley City, ND, 58072, USA
[2]Oregon State University, 1500 SW Jefferson Way, Corvallis, OR, 97331, USA
[3]African Institute of Mathematical Sciences, 6 Melrose Rd, Muizenberg, Cape Town, 7950

Abstract. Providing computing training to the next generation of physicists is the principal driver for a biannual multi-day training workshop hosted by the DUNE Computing Consortium. Materials are cast in a Software Carpentry's template, and topics have included storage space, data management, LArSoft, grid job submission and monitoring. Moreover, experts provide extended breakout sessions to demonstrate the fundamentals of the unique software used in HEP analysis. Each session uses live documents for real time correspondence, and are captured on Zoom; afterwards, videos are embedded on the corresponding web-pages for review. As a GitHub repository, shared editing of the learning modules is straightforward, and provides a trusted framework to extend to other training topics in the future. An overview of the tutorials as well as the machinery used, along with survey statistics and lessons learned is presented.

1 Introduction

The Deep Underground Neutrino Experiment (DUNE) is an international collaboration working to measure CP violation in neutrino oscillations, and simultaneously Supernova burst neutrinos. Additionally, the DUNE detectors will be sensitive to MeV scale solar neutrinos and will be used for rare and exotic BSM searches [1].

With a detector acceptance that spans multiple orders of magnitude in energy, a wider variety of analysis software prompts additional learning requirements for experimental high energy particle physicists who often arrive professionally with limited formal computing training. For these, DUNE Computing has been working to develop effective hands-on and well coordinated training events for the purpose of jump starting researcher's analysis work.

DUNE is not the first high energy physics (HEP) experiment to grapple with an organized approach to software training: ATLAS, Belle II, CMS are experienced contemporaries in training techniques [2] and via weekly meetings of the HEP Software Foundation (HSF), strategies and lessons learned are being shared [3]. So as to not reinvent the wheel, as a practice, we direct DUNE collaborators, be they graduates, post-docs, and tenured researchers to existing HSF training [4] that includes the Unix Shell, Git and GitHub, Python, and ROOT.

A goal for the DUNE computing tutorials has been to verify that new colleagues have access to computing resources, understand the basics of logging in from a terminal window, storage areas, running applications, code modifications, and submitting and monitoring batch jobs. An overview of the content and infrastructure of our computing tutorials follow.

*Presenter, e-mail: david.demuth@vcsu.edu

2 Basics Training for DUNE Computing

A small but dedicated squad of computing experts have contributed to the design and implementation of training materials focused on the basics of DUNE Computing. These DUNE-specific training events have been provided via multi-day workshops as a part of each collaboration meeting since 2016. Several dozen participants are typical for each training event, with the total number participants to date exceeding 400.

2.1 Curriculum Overview

The essentials for engaging in DUNE analysis work includes topics on data storage, data management, using *art* and LARSoft, modifying event analysis code, alongside job submission and monitoring [5]. A short synopsis of each basics lesson follows.

2.1.1 Data Storage

DUNE data volumes, commands, and tools to handle data are described in this lesson. Disk properties discussed include total size, available size, mount point, and device location. Handling data on grid accessible and interactive data volumes is demonstrated using commands such as `ifdh`, `xroot`, `xrdfs`, and `dcache`. Understanding data storage prompts good practice and minimizes the need to police usage.

2.1.2 Data Management

This learning module gives users an opportunity to observe experts who manage HEP data using SAM [6] and Rucio [7]. For example, elements of processed event data can be excerpted using `samweb` to list files associated to a particular run number and run type; `ups`, `mrb`, and `cvmfs` are also highlighted data management command line actions.

2.1.3 art and LArSoft

art is a generic software framwork used by many Fermilab experiments. LArSoft is a specialized toolkit for simulating and reconstructing data from Liquid Argon time projection chambers [8]. LArSoft is used by several experiments, one of which is DUNE, others include MicroBooNE, LArIAT, and ArgoNeut. Analysis setup configurations are demonstrated, and the `ups` command is used to probe data attributes, for example verifying the GEANT4 version being used in an analysis. The command `fhicl-dump` provides event data counts, among other useful information, and *art* produces files that are used by `ROOT` for analysis.

Learning to use *art* and LArSoft tend to be challenging, and this tutorial is considered introductory. A lesson on how to code for better efficiency provides support for new or rusty C++ users. An Expert-in-the-Room session is appended as a hands-on activity to coach students on the intricacies of the software.

2.1.4 Job Submission and Monitoring

This training module seeks to develop a respectful job submission acumen for new users who might lack an understanding of the impact runaway jobs can have on data storage and network bandwidth. Instruction focuses on using the `jobsub` batch submit command to control submission of jobs to local processing, OSG and WLCG, includes hints on debugging code, and offers a demonstration of best practices. In an associated "Job Monitoring" module, a sophisticated job monitoring web utility POMS is described and modeled [9].

3 Training Logistics

In what has become a standard for HEP meetings, Indico is used for event registration, communication, and event archiving. In advance, registrants are encouraged to work through the HSF's Unix Shell lesson [10]. Crucial to the goal for learning effectiveness is the ability to actively follow the instructor's live coding examples. To this end, we require a pre-training activity [11] that takes trainees through the steps needed to ensure a valid user account and access to DUNE interactive computing resources at Fermilab and/or CERN.

Live coding, quizzes, Expert-in-the-Room sessions ensure learning is active and accessible. Mentors are selected from previous training events and from experienced collaborators, their charge is to remain attentive to students throughout the event – and ideally beyond. Each session is offered in a face-to-face classroom setting and is simultaneously delivered via Zoom. Afterwards, Zoom-captured videos are lightly edited, captioned, and then embedded in lesson materials for asynchronous access.

3.1 Lesson Infrastructure

Colleagues at The Carpentries [12], who are steeped in developing coding and data science skills in researchers worldwide, have created pedagogically sound, and well vetted lessons for a variety of topics. The Software Carpentry (SWC) [13] offer an aesthetic and functional lesson template [14] that DUNE lessons utilize – notably, the author's attendance of the HSF's CI/CD workshop triggered DUNE's usage of these templates [16].

The infrastructure to develop lessons is provided in the SWC toolkit. A lesson template [15] is imported (not cloned) as a new DUNE GitHub repository, configuration for the event is made when editing a `yml` configuration file alongside lesson content which are located in the `_episodes` subdirectory; lessons content is assigned to selected instructors who author individual episodes, e.g., `02-storage-spaces.md`. GitHub Desktop is used to manage the repository locally versus raw editing markdown on GitHub. Monitoring edits in a localhost browser uses a Ruby/Jekyll engine [17] that is encouraged and optionally installed by lesson editors. Lessons are rendered elegantly on the web via the GitHub `gh-pages` framework.

3.2 Lesson Deployment

End users have access to both the raw GitHub hosted code, and the rendered `.io` site where navigation between lessons is seamless. Instructors benefit by having their full lesson visible to both them and their student. Lesson materials sided by a terminal window is the preferred live coding technique. Complex bash commands are easily copied from code blocks, and pasted by the student from the lesson page into their own terminal window for a time saving, yet effective practice. Quiz blocks have the feature of hidden drop-down answers, used by instructors to discuss and contrast possible solutions.

3.3 Support

During each lesson, a live and temporarily globally editable Google document (livedoc) is used for real-time questioning by students, with a cohort of experts engaged in answering those questions. Afterwards, the livedoc becomes an archive of the event for self-study afterwards. An important element of the tutorials is the relationships that form between the students with other students, students and instructors, mentors, and organizers. Slack channels are used for asynchronous support, as does an FAQ site hosted on GitHub issues [18]. Long term monitoring of Slack channels can be time intensive, where those who contribute most might earn a badge of honor, for example, or be allocated some fraction of "shifter" effort required for active collaborators.

3.3.1 Documentation

Self-study of documentation is an important training step. Newly on-boarded and longstanding researchers access DUNE computing resources which are inventoried in a protected one-stop MediaWorks wiki with main topic areas: Organization & Partners, Computing Toolbox, Operations & Monitoring, Working Groups, Resources, Getting Started. Redmine and GitHub are used for code documentation. For unrecognized topics: the ABC DUNE glossary. A CHEP 2023 poster outlines DUNE's approach to providing informative MediaWiki pages that captures "all things computing" for DUNE [19].

3.4 Student Snapshot

In the analysis of a survey given for the May 2022 training (N=35), graduate students (51.4%) and postdocs (28.6%) together make up the majority of our training audience. It is exciting to see undergraduates (8.6%) are participating. Few (5.7%) are expert at large batch jobs. A strong majority know Bash, use GitHub, Python, C++, and CERN's ROOT. Over half have yet to start their physics analyses (51.6%) – it was this population who we focus the event registration communications. A healthy number of participants (52.9%) reported their use of GitHub for code backups.

In follow up conversations, instructors reported that they recognized several students were stumbling, and then would adapt their lesson delivery to "meet them where they are."

3.5 Lessons Learned

The Carpentry lesson template is elegant, functional, and practical for delivering hands-on learning materials. Building and maintaining the lesson site is a significant but straightforward effort. Pre-event homework that includes checking that students can access FNAL servers is a must. It is easy to be ambitious with the learning objectives for one half day introductions, two day, or three day events. Coffee breaks during training events are important for assimilation, and are best if left as free time, instead of orchestrating networking activities; while some might seek tutoring during breaks. Mentors are essential to ensure skill development and understanding is widely successful. Hybrid synchronous delivery of workshops is ideal. Zoom captures hosted on YouTube and subsequently added to lessons, enhance asynchronous access to materials, and improve the record of the event. Training events during collaboration meetings are convenient for many, but lesson development might conflict with the typical workload that leads up to those meetings. As training materials are identified, instructors appreciate instructional design support. From an effort perspective, training, support, and documentation are related activities.

4 Future Work

Future work includes extending our experiences with the basics to specialized training, for example, providing training on accessing and using a hardware database, which is central to operations, has been suggested. Another expressed area is on the computing frameworks topic, for example event processing and display utilizing GPU technologies. Developing QA/QC training for detector component installers is another example. As the volume of the training materials specific to DUNE evolves, training the trainer sessions are warranted. For each training event, implementing pre- and post-survey to measure and demonstrate learning gain and to refine impact is essential.

HEP computing includes numerous data management, reconstruction, simulation, visualization, analysis, and documentation software applications and systems. Steering collaborators to scheduled training events by the community at large is a good practice. Evolving DUNE specific software training to a broader HEP community is a stretch goal. Examples include the metacat data catalog and its integration with Rucio which are being developed for DUNE but attracting interest from other experiments.

5 Conclusions

DUNE Computing has developed multi-day training events to focus on the basics with an aim to jump start individual's simulation and reconstruction analyses. Expert instructors practice clear communication, produce materials that are straightforward, are patient with students, offer access for questions though livedocs, and long term support via Slack channels, GitHub, and email. Rich with multicultural heritages themselves, DUNE collaborators are conscientious when developing curriculum to ensure the innate diversity becomes a feature for learning. DUNE Computing is committed to training, recognizes that the materials developed for DUNE has pertinence for other experiments, and is working to contribute to the larger open science community. Tutorials, support, and documentation are interconnected activities.

Ultimately in these tutorials we aim to inspire next generation software developers who in fact evolve to mentor, instructor, and expert levels, and who remain committed for the multiple years that the DUNE Collaboration will be operating.

Acknowledgements: This document was prepared by the DUNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. This work was supported by CNPq, FAPERJ, FAPEG and FAPESP, Brazil; CFI, IPP and NSERC, Canada; CERN; MŠMT, Czech Republic; ERDF, H2020-EU and MSCA, European Union; CNRS/IN2P3 and CEA, France; INFN, Italy; FCT, Portugal; NRF, South Korea; CAM, Fundación "La Caixa", Junta de Andalucía-FEDER, MICINN, and Xunta de Galicia, Spain; SERI and SNSF, Switzerland; TÜBİTAK, Turkey; The Royal Society and UKRI/STFC, United Kingdom; DOE and NSF, United States of America.

References

[1] DUNE Collaboration, DUNE Offline Computing Conceptual Design Report (2022), arXiv:2210.15665 [physics.data-an], https://doi.org/10.48550/arXiv.2210.15665

[2] S. Malik, Software Training in HEP (2021), arXiv:2103.00659 [hep-ex], https://doi.org/10.48550/arXiv.2103.00659

[3] HEP Software Foundation, Weekly Coordination, https://indico.cern.ch/category/10294/

[4] HSF, Training Center, https://hepsoftwarefoundation.org/training/center.html

[5] DUNE Computing, DUNE Computing Tutorials for May 2023, https://dune.github.io/computing-basics/

[6] Sequential Access via Metadata, https://wiki.dunescience.org/wiki/SAM

[7] Barisits, M., Beermann, T., Berghaus, F. et al. Rucio: Scientific Data Management. Comput Softw Big Sci 3, 11 (2019). https://doi.org/10.1007/s41781-019-0026-3

[8] LArSoft, https://larsoft.org/important-concepts-in-larsoft/

[9] Production operations management system, https://cdcvs.fnal.gov/redmine/projects/prod_mgmt_db
[10] The Unix Shell, https://swcarpentry.github.io/shell-novice/
[11] DUNE Computing, https://dune.github.io/computing-basics/setup.html
[12] The Carpentries, https://carpentries.org/index.html
[13] Software Carpentry, https://software-carpentry.org
[14] The Carpentries, example lesson template, https://carpentries.github.io/lesson-example/
[15] The Carpentries, base lesson template, https://github.com/carpentries-incubator/template
[16] HSF Software Training Center, Continuous Integration/Continuous Development, https://hsf-training.github.io/hsf-training-cicd/index.html
[17] The Carpentries, Ruby/Jekyll setup, https://carpentries.github.io/lesson-example/setup.html
[18] DUNE Computing, Computing FAQ, https://github.com/DUNE/Computing-Projects/issues
[19] C. David, All Things Computing, DUNE, CHEP 2023, https://indico.jlab.org/event/459/contributions/11681/