DUNE Computing

• The physics realms of the Deep Underground Neutrino Experiment presents a unique challenge for data analysis and data processing software frameworks:
  - Precision neutrino oscillation measurements,
  - Searches for nucleon decay,
  - Sensitivity to nearby Supernova explosions, and more…

• Particle interactions will span timescales of nanoseconds to 100’s of seconds, where event energies and associated processing timelines require a wide arsenal of software tools.

• Moreover, data storage is cloud-based and worldwide, and require failsafe high bandwidth networks with strong security and monitoring protocols.

DUNE Computing

• With the first of four detector designs prototyped, blasting of the deep underground caverns underway at Lead, SD, detector installation anticipated for 2027, and science starting in 2029, computing system development has been vigorous.

• ArgoNeuT, and MicroBooNE are LArTPc experiments that have produced software that DUNE Computing has adopted to ensure efficient liquid argon detector designs, those modelling techniques are highly numerical, and sophisticated to use.

• Graduate students, post-docs, young and senior researchers are among the hundreds of collaborators added each year, with most requiring general and specific software training.

• While selected training can be outsourced to an evolving curricula provided by the HEP Software Foundation (HSF), some training remains specific to DUNE.
DUNE Trainings

- DUNE Computing first sponsored training in 2016, and since coordinated 10 events in various formats, with ~400 participants.
- Tutorials have focused on three topics:
  - Data storage and management,
  - LArSoft (software for LArTPCs),
  - Job submission and monitoring.
- The goal is to certify that new colleagues have access to DUNE computing resources and understand the basics of logging in, storage areas, running applications, code modifications, submitting and monitoring batch jobs.

The May 2021 training was offered as a three day event, each of four lecturers and two mentors doing the bulk of the work.
Training Logistics

• Event registration and communications are managed using Indico.
• Participants must verify their ability to use the Unix Shell.
• Pre-event homework is vital as it includes verification that participants can access the DUNE general purpose virtual machines, and ideally the corresponding CERN VM’s (LX+).
• Livecoding, quizzes, expert in the room sessions, and assigned mentors ensure the event as hands-on.
• Each session is delivered and captured via Zoom, then embedded into the Software Carpentries lesson framework (SWC) which is hosted at DUNE Computing’s GitHub site for review.

DUNE Computing Training May 2021 edition: Mission Setup

Objectives
• Get ready to do the tutorial
• Understand the authentication procedures
• Set up your environment for DUNE
• Do an exercise to help us check if it’s all good
• Get streaming and grid access

Requirements
You must be on the DUNE Collaboration member list and have a valid FNAL or CERN account. See the Indico Requirement page for more information.

1. Kerberos business

If you already are a kerberise-efficio, go to the next section. If not, we give you a little tour of it below.

What is it? Kerberos is a computer-network authentication protocol that works on the basis of tickets.

Why does FNAL use Kerberos? Fermilab uses Kerberos to implement strong authentication, so that no passwords go over the internet (if a hacker steals a ticket, it is only valid for a day).

How it works? Kerberos uses tickets to authenticate users. Tickets are made by the krbtix command, which asks for your kerberos password (info on kerberos password here). The krbtix command needs the password, encrypts it and sends it to the Key Distribution Centre (KDC) at FNAL. The Kerberos configuration file, which lists the KDCs, is stored in a file named krb5.conf. On Linux and Mac, it is located here:

Code

```
/etc/krb5.conf
```

If you do not have it, it creates. A FNAL template is available here for each OS (Linux, Mac, Windows). More explanations on this config file are available here if you’re curious.

To log in to a machine, you need to have a valid kerberos ticket. You don’t need to do this every time you login, only when your ticket is expired. Kerberos tickets last for 26 hours. To create your ticket:

Code

```
krbtix -f username@FNAL.GOV
```

In advance of the opening salvo, students must demonstrate an understanding of using the Unix shell to access secure VMs.
Lesson Development

The infrastructure to develop lessons is provided in the Software Carpentries framework:

- **A lesson template** is imported as a new DUNE GitHub repo, configuration via a _config.yml file, and main lesson content as markdown files (.md) located in _episodes/.
- GitHub Desktop is used to manage the repository locally.
- Raw editing of files on github.com is frowned on, editing .md files locally is encouraged.
- Viewing edits in a localhost browser uses a Ruby/Jekyll rendering engine.
- As edits are verified, lessons are pushed to the site’s main branch for access by multiple authors, and the public.
- Lessons are rendered elegantly on the web via GitHub Pages as a free service.
Lesson Deployment

End users see:

GitHub's gp-pages are rendered seamlessly, and for free.

Markdown language features:

- Simplified editing structure for web browsers,
- Bash script blocks easily copied and pasted in terminal windows,
- Drop down Quiz blocks that can be opened on demand.
Getting Help

As expert instructors work through a lesson plan, often live coding in an adjacent window, multiple methods are used to ensure near- and long-term support:

- Instructors encourage participants to pose questions via a shared and world editable Google Doc (Livedoc) which is monitored by mentors and other experts.
  - Each participant selects a unique color to improve clarity.
  - As questions are volleyed, experts will form a dialogue with individuals to reason solutions; often a second expert will contribute.
- Livedocs can be studied by others, solutions can be contrasted, and the Livedoc becomes a permanent educational resource.
- Slack channels are set up for each training event, where questions can be posed asynchronously, when questions receive attention from a cadre of experts.
- GitHub Issues has proven to be another method to field questions, provide mentorship, and ensure the novice become proficient in DUNE computing.

Livedocs, Slack, and GitHub Issues are used to assist DUNE colleagues with computing questions.
Graduate students and postdocs make up the majority of our training audience. It is exciting to see undergraduates are participating.

Few are familiar with large batch jobs.

A strong majority know bash, use GitHub, Python, C++, and Root!

Over half have yet to start their physics analyses, and these were who we focused on in the event registration communications.

A healthy number of participants use GitHub for code backups.

Below, one student provides feedback after participating in a training.

The FNAL grid job submission has always been a pain for us. It seems very cumbersome to run anything on grid. I found this fact from the oscillation analysis I am working on (some inherited job submission scripts) as well as from the past dune computing tutorials. We need set up so many softwares for LArSoft, dune tpc, mrb as well as things like xrootd, IFDH, SAM input, etc. This maybe very necessary and useful for experienced users, but it’s very hard for undergraduate/early graduate to learn. Part of these is due to me not wrapping my head around all the softwares so far used at DUNE (and no one is teaching us).
Lessons Learned

• SWC lesson template is elegant, functional, and practical for delivering hands-on learning materials.
• Pre-event homework that includes checking that students can access FNAL servers is a must.
• It’s easy to be ambitious with the material for one half day introductions, two day, or three day events.
• Coffee breaks during trainings are important for assimilation.
• Mentors are essential to ensure skill development and understanding.
• Hybrid synchronous delivery is ideal.
• Zoom captures on YouTube allow asynchronous access, and a record of the event.
• Other approaches to training, such as Hackathons, could be used to take lessons to a next step.
HEP Software Skill Development

- The list of skills that experimental HEP physicists use on a regular basis is lengthy and evolving.
- Mastering data analysis techniques is not specific to one experiment.
- Using HSF training materials such as Unix Shell is effective and efficient.
- DUNE Computing encourages participation in HSF workshops.

This excellent instructional model by the HSF introduces concepts of continuous integration and development using Github motivated DUNE Computing’s use of the SWC lesson templates.
Due to the vast suite of software used by the DUNE collaboration for the various analysis steps and sub-detectors, training is important, with some on an ad-hoc basis by physics and hardware groups, e.g. software systems which monitor QA/QC for installation of the far detector.
Future Work

- HEP experiments host a large number of collaborators, many have limited computer science training.
- The movement is for software to be reused by multiple experiments; common tools are anticipated.
- DUNE Computing’s training work thus far has been focused on a few essential topics.
- Use of the Carpentries templates provides a proof of concept for other topics which have wider appeal.

A rough estimate is that there are ~10K people in the HEP community to train every year, ranging from undergraduates to scientists.

Software Training in High Energy Physics, Michel H. Villanueva, Sudhir Malik, Meirin Oan Evans
Summary

Students and postdocs who join HEP experiments who have no formal software training but become proficient with structured training, mentorship, and peer support.

**DUNE Computing has developed multi-day training events** to focus on the basics:

- Understanding data storage and best practices for operating in a cloud infrastructure,
- Learn event reconstruction and analysis techniques in liquid argon experiments,
- Master batch submission practices in a HPC grid environment.

Our **aim is to jump start individual’s simulation and reconstruction work.**

Expert instructors practice clear communication, produce materials that are straightforward, are open and patient with students, offer access for questions through 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, not a bug.

**DUNE Computing is committed to software training,** recognizes that the materials developed for DUNE has pertinence for other experiments, and is working to contribute to the larger open science community.
CHEP 2023 Abstract

Providing computing training to the next generation of physicists is the principal driver for a biannual multi-day workshop hosted by the DUNE Computing Consortium. Materials are cast in the Software Carpentries templates, and to date topics have included storage space, data management, LArSoft, grid job submission and monitoring. Moreover, experts provide extended breakout sessions to demonstrate the intricacies of the unique software used in HEP analysis. Each workshop session uses live documents for real time correspondence, and are captured on Zoom; afterwards, videos are embedded on the corresponding webpages 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 machinery will be provided, post workshop statistics will be discussed, with lessons learned will be the focus of this presentation.