# Introduction to Fermilab Computing Environment

## 2021 Intensity Frontier Computing Summer School

Lisa Goodenough for the FIFE Team

17th June 2021

# Housekeeping First

- Meeting Agenda
- What this School IS
- What this School IS NOT - please keep in mind that we have participants from several different experiments/projects here (g-2, Dune, NOvA, SBND, …) so we cannot make everything specific to your experiment
- **Question Board**: if you have questions during any of the sessions (except this one), please post them on this Google doc: https://docs.google.com/document/d/1b_Gr4PSoz1FdZ8-Mf1TQ1MRwb3F0zNEH_nUcH8j6PJU/edit?usp=sharing I will keep an eye the Board and will interrupt the instructors with questions posted there.

# Welcome

- Welcome

- Who am I?
  - Member of Muon g-2 and Mu2e Collaborations
  - Deputy Head of FIFE Project
  - HEPCloud Developer

- What is FIFE (Fabrlc for Frontier Experiments)?
  - Provides collaborative scientific data processing solutions - tools and services- for Intensity Frontier experiments
  - Modular - experiments can take what they need, and as new tools from outside communities become available they can be incorporated
  - Supports job submission, data management and handling, database and dataset applications such as beam monitoring, conditions and hardware, and more

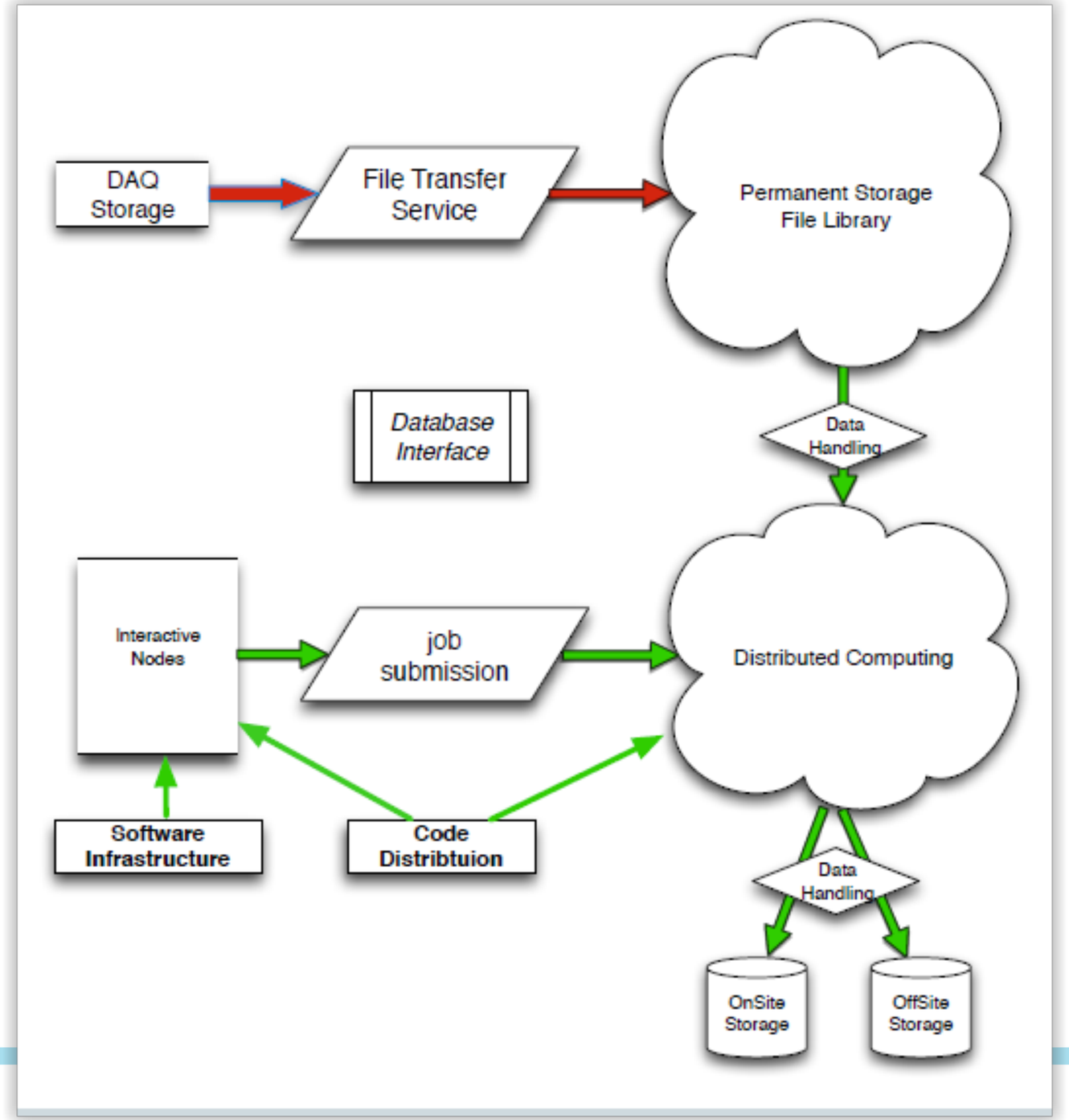
# My Goal

- Provide an overview of the tools and services used by experimentalists to do data processing and analysis

- Introduce names and terms that will come up in the presentations by the instructors over the course of the school

- Organization:
  - Computing Infrastructure
  - Scientific Software
  - Data Handling Tools

# Overview






Fermilab

# COMPUTING INFRASTRUCTURE

- Interactive Login Machines
- Distributed Computing (FermiGrid, OSG, Wilson Cluster, Cloud Computing Centers, HPC Centers,…)

# Interactive Login Machines

- GPCF (General Purpose Computing Facility) Interactive servers are generally provided as Virtual Machines running Linux called **gpvms**
- **Each experiment has its own gpvms** with names such as <experiment_name>gpvm#.fnal.gov (e.g. mu2egpvm02.fnal.gov)
- You should all have access to/accounts on gpvms belonging to the experiment(s) you work with - **login is via ssh with Fermilab kerberos credentials**
- You have a home area at /nashome/<initial>/<username> (the same home area is on all machines) and some disk space for data
- Same disks are mounted by all the interactive machines - i.e. my /nashome/g/goodenou directory is visible on all gpvms

# Distributed Computing

- FermiGrid - general purpose grid cluster, FermiGrid, that is shared by many experiments to run their physics jobs. Grid computing is a form of distributed computing in which multiple clusters of nodes work together to complete tasks. The grid determines which resources are free and uses those nodes to process the job

- Open Science Grid (OSG) - high-throughput grid consisting of clusters of computers at many different places

- Cloud Computing Centers, HPC Centers, …

# SCIENTIFIC SOFTWARE

- Frameworks
- Software Management - Version Control
- Software Management - Build Tools
- External Product Management - UPS/Spack

# Frameworks - art and LArSoft

- "***art*** is an event-processing framework developed and supported by the Fermilab Scientific Computing Division (SCD) to build physics programs by loading physics algorithms, provided as plug-in modules"

- *art* is a **suite of tools, libraries, and applications** for processing experimental data and simulated events

- Intensity Frontier experiments use *art* as the framework for their Offline software

- **LArSoft** is an *art*-based shared toolkit used by the LArTPC experiments

- *art* framework coordinates the processing of events by user-supplied pluggable modules that do simulation, reconstruction, filtering, and analysis tasks - modules are written in C++

# Frameworks - art and LArSoft

- "***art*** is an event-processing framework developed and supported by the Fermilab Scientific Computing Division (SCD) to build physics programs by loading physics algorithms, provided as plug-in modules"
- *art* is a **suite of tools, libraries, and applications** for processing experimental data and simulated events
- Intensity Frontier experiments use *art* as the framework for their Offline software
- **LArSoft** is an *art*-based shared toolkit used by the LArTPC experiments
- *art* framework coordinates the processing of events by user-supplied pluggable modules that do simulation, reconstruction, filtering, and analysis tasks - modules are written in C++
- **Tom Junk will be leading you through some of the details of *art* in his talk today**

# Software Management - Version Control

**Version control is the practice of tracking and managing changes to software code**

- Version control systems (VCS) keep track of every change made to code in a special kind of database
- Invaluable tool for software developers
  - aid in conflict resolution, which can occur when two developers modify the same piece of code at the same time
  - allow developers to more easily fix bugs by comparing different versions of the code, while minimizing disruption to other developers
  - enable teams to revert to a previous stable version of code in event of bugs
- For scientific collaborations, VC serves an additional purpose: allows for reproducibility of scientific results - data collection, processing and analysis can be done with the same version of the code even when the code has progressed
- Most common VCS in use: **Git (https://git-scm.com)** - lots of helpful documentation on web

# Software Management - Version Control

**Version control is the practice of tracking and managing changes to software code**

- Version control systems (VCS) keep track of every change made to code in a special kind of database
- Invaluable tool for software developers
  - aid in conflict resolution, which can occur when two developers modify the same piece of code at the same time
  - allow developers to more easily fix bugs by comparing different versions of the code, while minimizing disruption to other developers
  - enable teams to revert to a previous stable version of code in event of bugs
- For scientific collaborations, VC serves an additional purpose: allows for reproducibility of scientific results - data collection, processing and analysis can be done with the same version of the code even when the code has progressed
- Most common VCS in use: **Git (https://git-scm.com)** - lots of helpful documentation on web

**"The Missing Semester of Your CS Education" Git tutorial was recommended material for this course. If you haven't completed that tutorial I urge you to do so.**

# Software Management - Build Tools

- Cannot get too specific here - experiments use different tools (SCons, mrb, …) to build their proprietary software

- As external products, they are setup using UPS (see next slide)

- **mrb** is based on **make** and simplifies the building of multiple products pulled from separate repositories

- **SCons** uses Python scripts as "configuration files" for software builds

---

- **make** is a standard build tool that determines dependencies, build order, and issues the commands (uses `Makefile(s)` for configuration and construction)
- **cmake** is a tool with a simpler configuration language that writes all of the `Makefile(s)`
- **cetbuildtools** (a package developed at FNAL) provides convenient macros for `cmake` (used

# External Product Management - UPS (UNIX Product Support)

- Software support toolkit (developed at Fermilab) for management of external software products on local systems (includes art, Geant4, ROOT, gcc, …)
- See https://scisoft.fnal.gov for a full list of available products
- Provides a uniform interface for accessing all products on UNIX (or UNIX-like) systems via the **`setup`** command
- Products can be declared to depend on other products (specific versions, builds, and ‘flavors’) - a particular version of *art* depends on particular versions of ROOT, CLHEP, boost, and several other products


Fermilab

# External Product Management - UPS (UNIX Product Support)

- Products have
  - versions (coincide with version of software, e.g. Python 3.9.5)
  - flavors: distinguishes different OS's (SL6 vs SL7)
  - qualifiers: denote different builds coming from variations in compiler, optimization levels, required version of other products

- To setup the profile build of version 2_10_03a of *art* built for SL7 using GCC v6.4.0 with `-std=c++14`, `-std=gnu` (gfortran)

```
setup art v2_10_03a -f Linux64bit+3.10-2.17 -q+e15:+prof
```

- This not only sets up *art* but also all of the dependencies that go along with it

- Products can be unsetup using the **`unsetup`** command

# External Product Management - Spack

- UPS is nearing end-of-life - Spack has been picked to *replace* it
- Spack is a package manager for supercomputers, Linux, and macOS
- Supports combinatorial versioning (versions, flavors, build variations) like UPS
- Benefits for running code at supercomputing centers: good at forcing builds to use special optimization flags, lots of recipes for multithreading libraries, etc.
- Benefits for us: lots of documentation and videos
- See spack-infrastructure for Fermilab local Spack tools

# DATA HANDLING TOOLS

- Data Storage
- Data Transfer
- Data Management - SAM


Fermilab

# Data Storage

- Several categories of **disk** space at Fermilab:
  - **NAS home areas** - small quotas, available on all gpvms
  - **CVMFS** (CERN Virtual Machine File System) - distributed disk system used to provide pre-built releases of the experimental code and UPS products to all interactive nodes and grids
  - **dCache** - distributed disk system with a very large capacity used provide high-volume and high-throughput data interactively or in grid jobs - incorporates load balancing and optimization - designed to make a single file available to a single node
  - **stashCache** - provides the CVMFS interface with dCache storage - designed to make a few very large files available to all nodes during a grid job
- These break down into categories (persistent dCache, tape-backed dCache, …) - you will learn about these as you need to

# Data Storage - Tape

- Scientific Computing Division maintains a system of data tapes called **Enstore**
- Tapes are held in robotic libraries in Feynman Computing Center (FCC) and the Grid Computing Center (GCC) buildings
- Access to these files is through the **/pnfs file system**, which is part dCache
- Most important thing to remember - **you must prestage files from tape to disk prior to using them** - waiting jobs will block others from using shared resources

# Data Storage - Tape

- Scientific Computing Division maintains a system of data tapes called **Enstore**
- Tapes are held in robotic libraries in Feynman Computing Center (FCC) and the Grid Computing Center (GCC) buildings
- Access to these files is through the **/pnfs file system**, which is part dCache
- Most important thing to remember - **you must prestage files from tape to disk prior to using them** - waiting jobs will block others from using shared resources
- **Mike Kirby will be leading you through some of the details of Data Storage in his talk tomorrow**

# Data Management - SAM (Sequential Access to Metadata)

- Fermilab product containing **databases and servers**, designed to help manage large datasets or files

- Contains several parts
  - a database of **metadata** for each file
  - a database of **locations for files**, usually in dCache
  - servers called **SAM stations** which help manage files
  - SAM stations providing file names and locations to a grid job
  - job submission and cache **management features**

# Data Management - SAM (Sequential Access to Metadata)

- Fermilab product containing **databases and servers**, designed to help manage large datasets or files

- Contains several parts
  - a database of **metadata** for each file
  - a database of **locations for files**, usually in dCache
  - servers called **SAM stations** which help manage files
  - SAM stations providing file names and locations to a grid job
  - job submission and cache **management features**

- **Tammy Walton will be leading you through some of the details of SAM in her talk tomorrow**

# Conclusion

- Data transfer (using ifdh and xrootd) will be covered tomorrow
- Job submission and Monitoring Tools (jobsub, POMS,…) will be covered on Monday
- Please feel free to send me an email if you have any questions
- Thank you for attending!