MicroBooNE FIFE Experience

FIFE Workshop
June 1, 2015

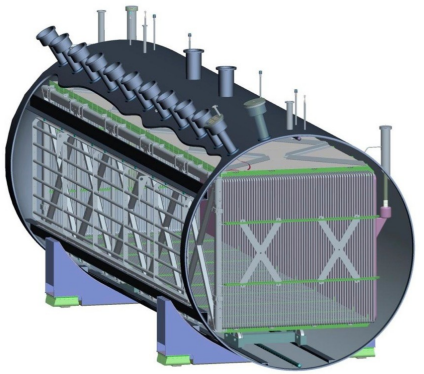
H. Greenlee

Outline

- MicroBooNE overview and status.
- Offline software environment (ups products).
- Frameworks (art, larsoft).
- Software development environment (mrb).
- Software releases and distributions (Jenkins, cvmfs).
- Batch jobs and workflow (jobsub, larbatch, pubs).
- Data Handling (sam, dCache).
- Databases.

MicroBooNE Overview and Status

- MicroBooNE is a large liquid argon time projection chamber (LArTPC) in located in the booster neutrino beam.
- Current status.
  - Cool down is taking place now.
  - LAr fill and HV ramp expected in June (expected to start taking physics data some time around end of June).
  - Commissioning with cosmic ray data will take place during summer shutdown (July-Sept.).



Offline Software Environment

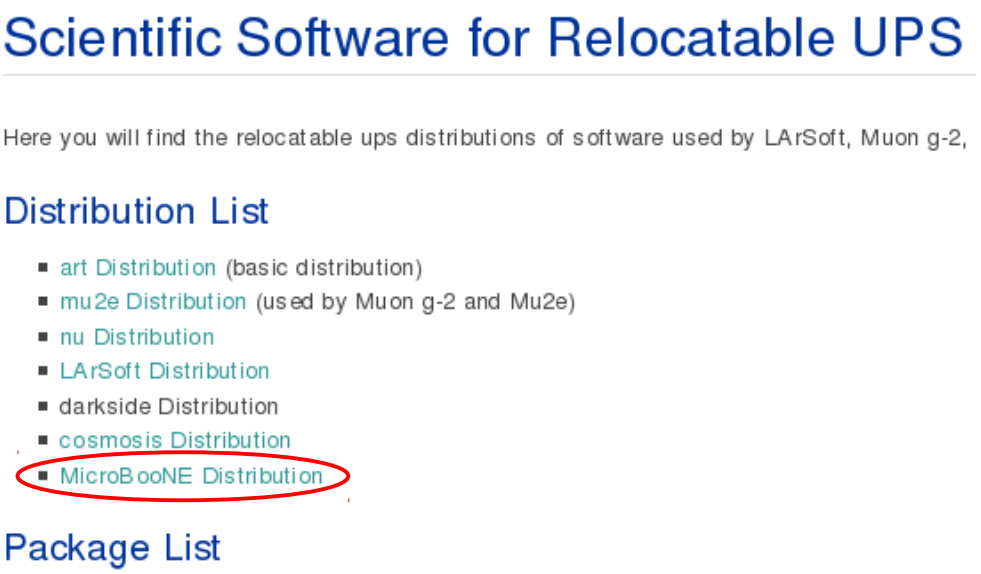
- Run-time environment based on three-tiered relocatable ups system.
  - Common products (/grid/fermiapp/products/common).
    - jobsub_client, sam_web_client, totalview.
  - Larsoft products (/grid/fermiapp/products/larsoft).
    - Art suite: art, genie, geant4, root, etc.
    - Lasoft proper (about ten separate ups products).
    - ifdhc, ifdh_art.
    - Database: postgres_client, psycopg2.
    - Others.
  - Microboone products (/grid/fermiapp/products/uboone).
    - Uboonecode (microboone-specific code that depends on larsoft).
    - Ubutil (microboone-specific scripts and tools).

Software Environment Distribution

- All three tiers of ups products are currently being distributed as SLF6 and MacOS (OSX 10.9, 10.10) binaries.
  - Most ups products (including all larsoft and microboone ups products) are available for download as relocatable ups tarballs from the scisoft products server.
    - Of the products we commonly use outside of Fermilab, only sam_web_client is not available from scisoft, and needs to be installed using upd (why is that?).
  - All three tiers of ups products are available using cvmfs from oasis.opensciencegrid.org server.

Scisoft Products Server

- Manifests of the full MicroBooNE software environment are available for download on scisoft home page.
  - Allows installation of full microboone environment (except sam_web_client) and all dependent ups products with a single command.



Larsoft

- Microboone is a member of the larsoft project.
  - Larsoft code base shared by multiple LAr TPC experiments: microboone, argoneut, dune, sbnd, lariat.
  - The majority of microboone's main simulation and reconstruction software is actually part of larsoft.
  - Larsoft is organized as nine packages.
    - Each package has its own redmine, git repository, and ups product.
  - Microboone currently has one additional package (redmine + git repository + ups product), that is not part of larsoft, for microboone-specific larsoft code, called "uboonecode."
- Larsoft team: Erica Snider, Gianluca Petrillo, Lynn Garren, Saba Sehrish.
- Larsoft uses the art framework.

MicroBooNE/Larsoft Development Environment

- MicroBooNE and larsoft code is stored in git repositories attached to redmine packages.
- Software builds, for development or distribution, are done using mrb (multi-repository build), which is built on top of cmake and cetbuildtools.
  - Mrb conveniently packages what it builds into relocatable ups products.
  - In my experience, hardly anyone really understands cetbuildtools. Maybe SCD could have a training seminar.

Jenkins Build System

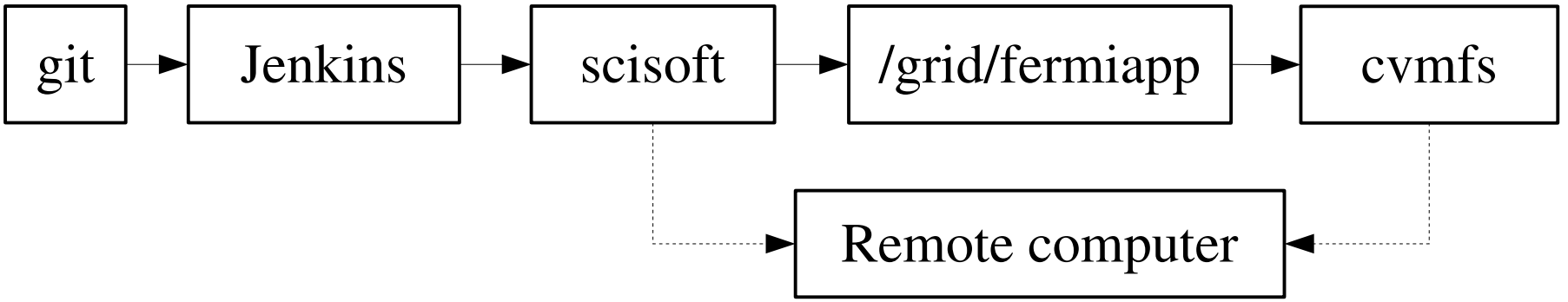
- Larsoft and uboonecode builds are done using Jenkins, a web-based multiarchitecture build system.
  - http://buildmaster.fnal.gov
  - Jenkins pulls code directly from git and produces relocatable ups tarballs and scisoft manifests.

Project uboone-release-build-mac

build uboonecode and ubutil

Configuration Matrix	debug	prof
OSX-10.10		
OSX-10.9		
SLF6		

- Release distribution flow.



Some Thoughts on Supporting MacOS

- Art, larsoft, and various experiments have been moving toward increasing support for MacOS.
- This effort to increase MacOS support has largely been successful, as microboone have almost the entire offline environment available as precompiled binaries for MacOS. People who like developing on their Mac laptops are mostly happy.
- There are a few issues.
  - A few bits are missing (notably ifdh).
  - Fermilab supported MacOS infrastructure is minimal-to-nonexistent.
    - Jenkins build slaves are few and weak compared to SLF build slaves.
    - No publicly available MacOS login or development nodes.

The MacOS CVMFS Cmake Problem

- Parallel builds in MacOS (“make -jN” or “mrb -jN”, with N>1) out of cvmfs always fail with the following error:

```
/oasis.opensciencegrid.org/fermilab/products/larsoft/cmake/v3_2_1/Darwin64bit+13/bin/cmake:
cannot execute binary file
```

  - I have seen this error on Jenkins build slaves, as well as my own laptop.
  - Service desk ticket open, no resolution.

Workflow Tools I (Larbatch)

- Redmine project larbatch, aka LAr Batch Tools, w/ git repo + wiki.
- Ups product larbatch is included in larsoft build, but is not a dependency of larsoft.
- Now being used by several larsoft experiments.
- Features.
  - Top level python script/module project.py.
  - Standard batch worker script condor_lar.sh (runs lar executable).
  - Batch submission (using jobsub).
  - Validation of results.
  - Makeup and resubmission of failed jobs.
  - Multistage projects.
  - Handles SAM interactions for reading and writing.
  - Command line, gui, python api.

Project.py GUI

File View Project Stage Output Batch SAM-art SAM-ana Help

XML Path: test5.xml

Project: prodgenie_numi_nu_cosmic_uboone

	Output						Check / CheckAna / Fetchlog	Batch Jobs	Submit / Makeup / Update		
Stage	Exists?	Art Files	Events	Ana Files	Errors	Missing	Idle	Running	Held	Other
gen	Yes	10	100	10	0	0	0	0	0	0
g4	Yes	10	100	10	0	0	0	0	0	0
detsim	Yes	10	100	10	0	0	0	0	0	0
reco1	Yes	10	100	10	0	0	0	0	0	0
reco2	Yes	10	100	10	0	0	0	0	0	0
mergeana	No	0	0	0	0	0	0	0	0	0

```
Checking directory /uboone/data/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
Checking root files in directory /pnfs/uboone/scratch/users/greenlee/devel/v04_08_02/reco2/prodgenie_numi_nu_cosmic_uboone/1865164
100 total good events.
10 total good root files.
10 total good histogram files.
0 processes with errors.
0 missing files.
```

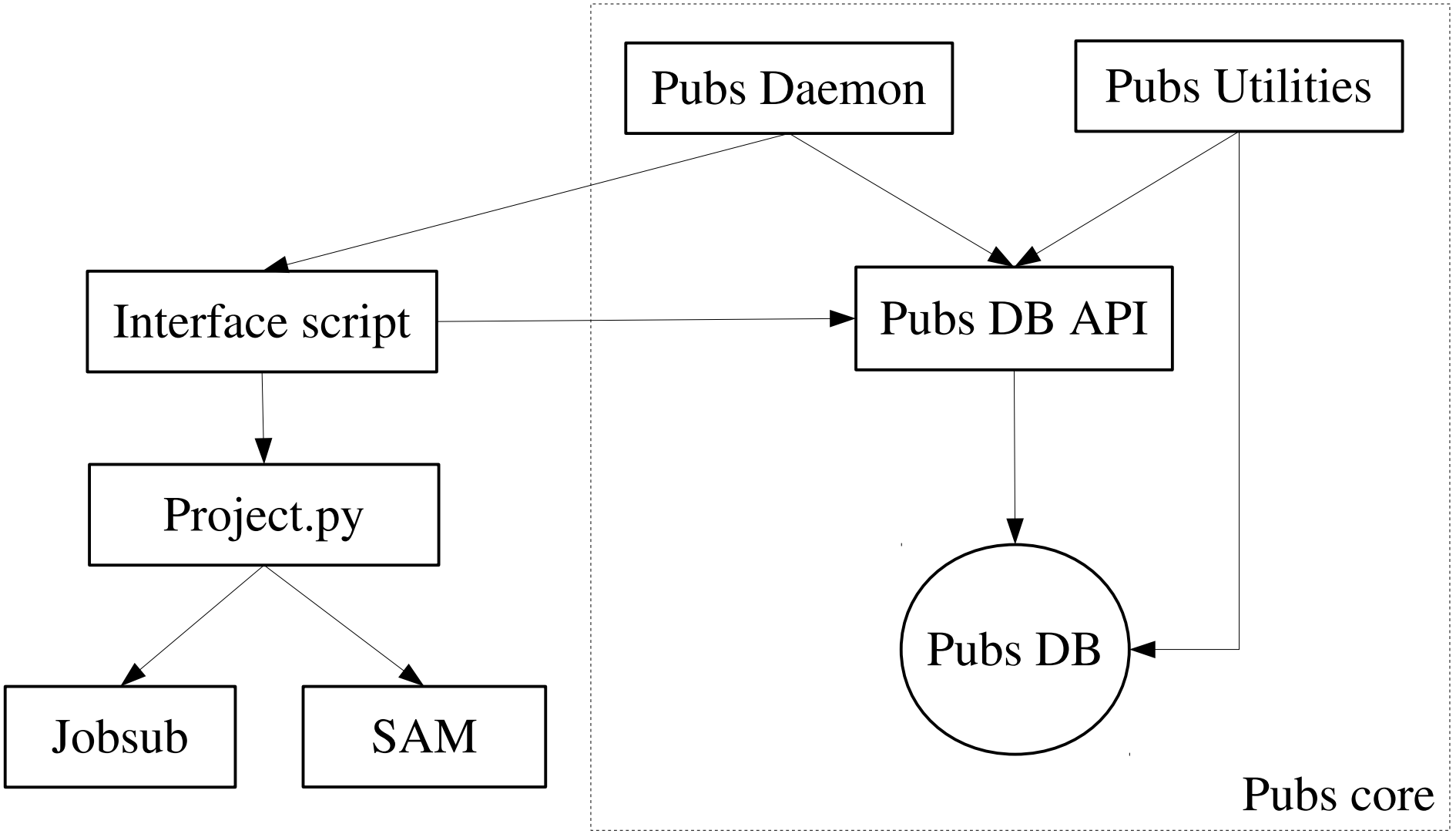
Workflow Tools II (Pubs)

- Pubs is a fully automated “fire and forget” production system being used by MicroBooNE.
- Adapted from a mature system used by Double Chooz (Kazu Terao).
- Pubs core functionality.
  - Production database & schema (postgres database).
    - Database records status of each production unit (e.g. file).
  - Database python api.
  - Stand alone utilities (e.g. configuration utilities).
  - Daemon.
  - Web interface.
    - However, the Double Chooz web interface has not been deployed here because of Fermilab security rules.

Workflow Tools II (Pubs, cont.)

- Pubs requires the use to supply a project interface script.
  - Invoked by pubs daemon.
  - Batch and data handling system interface.
  - Provides state machine logic.
  - In microboone, we can currently envision three such interface scripts.
    - Monte Carlo simulation + reconstruction processing chain.
    - Data reconstruction processing chain.
    - Binary to art format raw data conversion (“swizzling”).

Pubs Dependency Diagram



Data Handling

- Microboone data handling is based on standard Fermilab SAM system.
  - SAM file catalog.
  - Enstore tape storage.
  - File transfer service (FTS) enstore upload drop box.
- We have integrated sam into our art framework (ifdh_art) and work flow.
  - Generating metadata.
  - Storing metadata in sam database.
  - Uploading files to enstore.
  - Reading data using sam.

dCache

- During this year's SCPMT process, MicroBooNE was told not to expect more bluearc disk space for data storage.
  - But to use dCache instead.
- After a rough start last year, after dCache software upgrade and SLF6 migration, we successfully transitioned to using dCache scratch storage in most situations.
- We still have an unmet need for non-volatile, non-tape-backed disk storage.
  - At this year's SCPMT, we were told that there could be a possibility of using dCache in this way, but hasn't happened yet.

Databases

- Online databases (hosted at LArTF, managed by MicroBooNE).
  - Configuration (Trigger + DAQ).
  - Conditions (Slow controls & monitoring).
  - Coordination (PUBS, for online production, aka “swizzling”).
- Offline databases (managed by SCD).
  - Offline replicas of online databases.
  - Connections (electronic channel map & cabling).
  - Calibration (w/ IFDB http proxy server).
  - Coordination (PUBS, for offline production – not replica).

Summary

- Microboone has a stable offline software development and run-time environment based on larsoft, mrb, ups, etc.
- Software binary distribution is via scisoft and cvmfs.
- Microboone is using sam for data handling and dCache and enstore for data storage.
- Data production.
  - Currently mainly on FermiGrid.
  - Tested on OSG, but still limited by excessive memory use.