MicroBooNE Computing Overview

FIFE Workshop
June 16, 2014

H. Greenlee

Outline

- Experiment status.
- Computing Hardware
- Offline software environment.
- Frameworks (art & larsoft).
- Data Handling.
- Databases.
- Data Production.
- Collaboration tools.

Experiment Status

- LAr cryostat was welded shut last month.
- Cryostat is scheduled to move from DAB to LArTF on June 23 (one week from today).
- Commissioning is expected to take ~six months.
  - Microboone is expecting to be ready for beam (cryostat filled with LAr and full HV) around Dec. 2014.
  - It will be possible to take cosmic ray or parasitic data at partial HV earler than that.

Computing Hardware

- Six login vms, uboonegpvm01-06 (4 cpus, 12 TB).
  - Three slf5, three slf6 currently (will migrate to slf6 soon).
- 500 batch FermiGrid batch nodes.
  - Also using opportunistic slots on FermiGrid.
- Online and daq systems (administered by FEF).

Offline Software Environment

- Run-time environment based on three-tiered relocatable ups system.
  - Common products (/grid/fermiapp/products/common).
  - Larsoft products (/grid/fermiapp/products/larsoft).
    - Includes art suite, geant4, root, etc., as well as larsoft proper.
  - Microboone products (/grid/fermiapp/products/uboone).
    - Currently contains one ups product that we (microboone) actively maintain called "uboonecode."
- Development environment based on mrb (multi-repository build system), which runs on top of cmake.
  - Mrb is tightly integrated with ups.

Software Environment Distribution

- All three tiers of ups products are currently being distributed as slf5 and slf6 binaries via cvmfs at oasis.opensciencegrid.org.
  - With exception of some genie flux files, that are only available locally at Fermilab on bluearc disk.
- Larsoft and its dependent products (art suite) are now supported on MacOS.
  - Distribution is source only (so far?).
  - Some people have successfully run microboone offline software on their macs.

Larsoft

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

Larsoft Dependencies

- Larsoft dependent products.
  - Art framework & art suite.
  - Geant4.
  - Genie (main neutrino event generator).
  - Root.
    - Art is tightly integrated with root, including root i/o.

Larsoft Development Environment and Builds

- Larsoft recently migrated away from a development environment based on SRT (SoftRelTools) build system and svn repository, to an environment based MRB (multi-repository build) and git.
- The larsoft team is responsible for building larsoft slf5 and slf6 binaries, and uploading larsoft ups products to /grid/fermiapp and cvmfs.
- We (microboone) build uboonecode binaries and upload to /grid/fermiapp and cvmfs.

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 successfully 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.

Databases

- Slow controls monitoring database (online).
- Electronics channel map database (online).
- Run configuration database (online).
- Production database (planned, for online and offline, data and mc).
- Calibration database.
  - This is the only database that will need to be accessed from grid workers.
  - It will be served via http, (with Igor M.'s help).

Data Production

- So far, all microboone mc production jobs have been run on FermiGrid (dedicated and opportunistic slots).
  - So far, we have run four major MC productions.
- OSG
  - Software environment is OSG-ready (full binary environment is available via oasis cvmfs distributino).
  - We have successfully run test jobs on OSG using a modified workflow.
- Cloud
  - Also testing on FermiCloud and commercial clouds.

The Memory Problem

- Currently, one of the main problems we face when trying to run on the grid (FermiGrid, and especially OSG) is that our jobs sometimes crash because they use too much memory.
  - FermiGrid max VM 4 GB.
  - OSG max VM 2 GB.
- Solving the memory problem looks like it will require a major campaign and help from computing experts.
  - We are already working closely with art and larsoft teams to identify ways to reduce memory.
  - The memory campaign would certainly benefit from more effort and more help from SCD.

dCache / Xrootd

- We would like to migrate away from using bluearc and toward using dCache scratch for storing temporary files during production.
- Currently we are being limited by the following problem.
  - Nfs access to dCache scratch doesn't work (we now know).
  - We also haven't succeeded in getting streaming access using xrootd to work.
    - Service ticket has been open for a while.

Workflow

- Microboone has a fairly simple set of workflow scripts.
  - Standard batch worker script.
  - Job submission script.
- Workflow features.
  - Job submission to FermiGrid via jobsub_tools.
    - Will migrate to jobsub_client soon (including submission to OSG).
  - Validation and error recovery.
  - Handles all sam interactions.
    - Storing metadata and files.
    - Defining sam datasets.
    - Setting up sam projects for reading.

Collaboration Tools

- Electronic logbook.
- Docdb – for documents and meeting agendas.
- Redmines.
  - Larsoft and uboonecode (git repos).
  - Legacy larsoft-svn and ubooneoffline (svn repos).
  - Uboone DAQ.

Summary

- Microboone has a stable offline software development and run-time environment based on larsoft, mrb, ups, etc.
- Software binary distribution is via cvmfs.
  - No solution as yet for distributing genie flux files.
  - No binaries for MacOS.
- Microboone is using sam for data handling.
- Data production.
  - Currently mainly on FermiGrid.
  - Tested on OSG, but still limited by excessive memory use.
- Using standard collaboration tools docdb and redmine.