LBNE Software Installation System Status and Issues

Brett Viren

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

October 4, 2013

LBNE Software Installation Problem Space

- ∼40 externals, ∼5 LBNE sub-suites
  - build tools: CMake and bleeding edge GCC (!)
  - large HEP packages (Geant4, ROOT, GENIE)
  - Python, Boost both PostgreSQL and MySQL, gccxml, ...
  - A large framework: art + larsoft (+artdaq for online)
  - LBNE: FD/ND/Beam/Fast simulation, detector reconstruction projects
- Multi-national, distributed collaboration
- Multiple platforms: at least Sci. Linux, Deb/Ubu, Mac OS X
- Require, roughly in order of decreasing importance:
  1. Control versions independently from other experiments.
  2. Facilitate any collaborator building all relevant software from pristine sources on all supported platforms.
  3. Provide pre-built binaries.

LBNE Software Installation System

"worch = waf + orchestration"

- Meta-build system using a package's native build mechanisms (eg. autoconf, cmake, etc)
- Flexible (meta) configuration system
  - installation directory layout patterns
  - manages package versioning and overall build versioning
  - organization: groups of packages, suites of groups
  - express inter-package dependencies
  - details of build flags, environment.
- fine grained, sub-package parallel building (beyond "`make -jN`")
- Installation requires downloading a single file and running two commands.
- Being used outside of LBNE (ORKA and collaborating with ATLAS)

Development at: `https://github.com/brettviren/worch`

User Environment with "worch"

- Unified installation and environment configuration file set
- Multiple environment setup systems supported. So far:

  UPS FNAL-centric system for user environment setup
  - Used for user environment setup on fnal.gov machines
  - Somewhat cryptic configuration language
  - Sprinkled dependencies in FNAL installation "system"
  - Essentially is required to build Art.

  modules Environment modules (`http://modules.sf.net/`
  - Free Software, community supported
  - Uses a "real" language (well, at least TCL)
  - Already in use by some LBNE collaborators (PDSF)

Worch does not require a user environment system to build. They are added as just another build "feature".
However....

The problem with UPS and Art

- Art is the de facto software framework for LBNE offline (and online/DAQ?) at this time.
- It is built with CMake. That's great, however:
- Its CMake files require UPS.
- This greatly hampers building it outside of FNAL
- Building Art, itself, has posed more problems than all other externals combined.

Possible solutions:

- Rewrite Art's native build system to be independent from UPS
  - Art developers are not interested
  - Large effort for an outsider, always needing to chase differences
- "Fake" UPS and Art is non-the-wiser.
  - Probably more effort than the next solution:
- Provide for UPS to be used by worch installation elements
  - **This is the approach I'm talking now.**

Expected Time Line

- Worch "works" now.
  - It builds non-art, LBNE sub-suites.
  - It builds ORKA's Geant4/ROOT simulation/analysis application
- Building Art and Larsoft will take more effort.
- If this can't be done in 1-2 months I may give up.
  - If I give up, I fully expect that we must rely on FNAL to provide binary-only.
  - We loose the ability to just fix things ourselves.
  - A single person is currently doing this spread across Nova, Mu2e, Darkside50 and other FNAL neutrino experiments. LBNE would need to compete for attention.

Of course, help wanted.

- Try building non-art suites.
- Try new platforms (so far SL6 and Debian)
- Get involved in solving "The Art/UPS problem"
- Help incorporate building LArSoft (its current build system and repository location are changing and I've not "gone there" yet).