# Progress in Hall D Offline Software, 2015

GlueX-Doc 2921-v1

Mark Ito, David Lawrence, Curtis Meyer

February 11, 2016

We summarize new efforts in GlueX offline software development in the year just past.

## 1 Conversion from GEANT 3 to Geant4

Work continues on converting our GEANT 3 implementation of detector simulation to Geant4. The geometry definition complete in the new system. Here an XML-based specification is read in at run time and appropriate classes instantiated. Work continues on implementing detector hit generation. One of the main benefits to the new system will be the ability to run multi-threaded. Scaling of execution speed with threads has been demonstrated.

There are two implementations at present: one for the standard GlueX program (HDGeant4)[1] and one for the Charged Pion Polarizability Experiment (CPPsim)[2].

## 2 Adoption of SWIF

We have incorporated use of Scientific Computing's system for managing collections of related farm jobs, SWIF, into all of our large-scale computing projects. These include the bi-weekly trains doing reconstruction of recent data, simulation studies, and data challenges.

## 3 Electromagnetic Background Simulation

We have developed a method for incorporating a separately generated library of out-of-time electromagnetic background events into a sample of simulated physics events. This leverages the existing background generator, which takes into account a coherent photon beam and beam collimation to produce out of time hits. This method saves time by avoiding generation of E&M background on an event-by-event basis.

## 4 EventStore

Work continues on implementing EventStore (from CLEO) into GlueX. The system uses sets of independent event indices to allow access to a common reconstructed data set based on criteria that vary from index to index. This function is usually done by skimming data sets to produce specialized streams in independent data sets, but that method often results in many events contributing to multiple streams with a net increase in the size of the reconstructed data set. Compact reconstructed data, beyond being efficiently stored, is easier to distribute to off-site institutions.

# 5 Conversion to Git

This past summer we transitioned from using Subversion for source code version control to using Git. We had a program of explaining the new system to the collaboration, designing the style of usage we would adopt, and providing documentation to help the new users. There has been marked improvement in communication about changes to the code using the tools available via GitHub[3]. This completes a response to a recommendation from one of the Software Reviews.

We have also implemented a system where pull requests on GitHub generate a build at JLab using the branch proposed in the pull request. We can now judge whether a change will compile and link before it is merged into the master branch.

# 6 Spring 2015 Simulations Complete

We completed a set of simulations to support analysis of data taken in the Spring of 2015. Thirty thousand jobs were run successfully on the farm for this effort.

# 7 Software Distribution and Building

A system for managing our various software packages, both those externally provided and those developed in-house, was enhanced and documented during the year. The goal is to insulate the user from the need to the master details of building each of several software packages as well as from the details of setting up a working environment. Multiple versions of each of several packages can be maintained simultaneously. Particular combinations of package versions can be specified succinctly in an XML configuration file and this file can be used both to guide a complete build of all needed packages and to set up the shell environment to use the resulting build.

In a parallel development, one of our members has been working on a system focused on supporting multiple builds of particular packages to compare results among them. The system is used to create binary distributions of all packages needed to run and develop GlueX software.

# 8 Calibration Challenge

Work is in progress on consolidating and automatizing all known calibration activities. Work will proceed in several passes, each one dependent on the results of the previous pass. The goal is production of calibration constants soon after experimental data is taken. A framework has been developed and several collaborators have started contributing the working pieces in the form of run-time-loaded plugins.

The calibration effort in general is going well with all detector systems nearly at design resolution, if not better.

# 9 Clang-based C++ Code Analyzer

We are now doing nightly generation of reports on questionable passages of our C++ code base using the Clang/LLVM compiler suite. A web page is generated; problem areas are displayed and explanations for why these lines of code were flagged are given.

# 10 Upcoming Data Taking Run

We are planning to have the software components in place to be able to calibrate and process all data during the Spring 2016 run.

# References


[1] https://github.com/rjones30/HDGeant4

[2] https://halldsvn.jlab.org/repos/trunk/Experiments/PionPolarizability/src/CPPsim

[3] https://github.com/orgs/JeffersonLab/teams/gluex/repositories