



Contribution ID: 220

Type: Oral

Experiments Toward a Modern Analysis Environment: Functional Programming Style, Scriptless, Continuous Integration, and Everything in Source Control

Thursday 21 January 2016 16:10 (25 minutes)

A modern high energy physics analysis code is complex. As it has for decades, it must handle high speed data I/O, corrections to physics objects applied at the last minute, and multi-pass scans to calculate some corrections. More recently an analysis has to regularly accommodate multi-100 GB dataset sizes, multi-variate signal/background separation techniques, larger collaborative teams, and reproducibility and data preservation requirements. The result is often a series of scripts and separate programs stitched together by hand or automated by small driver programs scattered around an analysis team's working directory and disks. Worse, the code is often much harder to read and understand because most of it is dealing with these requirements, not with the physics. This paper describes a framework that is built around the functional and declarative features of the C# language and its Language Integrated Query (LINQ) extensions to declare an analysis. The framework uses language tools to convert the analysis into C++ and runs ROOT or PROOF as a backend to determine the results. This gives the analyzer the full power of an object-oriented programming language to put together the analysis and at the same time the speed of C++ for the analysis loop. The tool allows one to incorporate C++ algorithms written for ROOT by others. A by-product of the design is the ability to cache results between runs, dramatically reducing the cost of adding one-more-plot and also to keep a complete record associated with each plot for, to aid with data preservation and log-book annotation. The code is mature enough to have been used in ATLAS analyses. The package is open source and available on the open source site GitHub. Recent improvements include the ability to run jobs on the GRID and access GRID datasets as a natural part of the analysis code, further tools to help with data preservation, and a start towards incorporating tools like TMVA, the multivariate analysis package in ROOT, into the code.

Primary author: WATTS, Gordon (University of Washington (US))

Presenter: WATTS, Gordon (University of Washington (US))

Session Classification: Track 1

Track Classification: Computing Technology for Physics Research