Contribution ID: **221** Type: **Oral**

Go-HEP: writing concurrent software with ease and Go

Thursday, August 24, 2017 4:45 PM (20 minutes)

High Energy and Nuclear Physics (HENP) libraries are now required to be more and more multi-thread-safe, if not multi-thread-friendly and multi-threaded.
This is usually done using the new constructs and library components offered by the C++11 and C++14 standards.
These components are however quite low-level (threads, mutexes, locks, …) and hard to use and compose, or easy to misuse.

However, Go – a somewhat new language – provides a set of better building blocks for tackling concurrency: goroutines and channels.
This language is now used by the whole cloud industry: docker/moby, rkt, kubernetes are obvious flagships for Go.

But to be able to perform any meaningful physics analysis, one needs a set of basic libraries (matrix operations, linear algebra, plotting, I/O, …)

We present Go-HEP, a set of packages to easily write concurrent software to interface with legacy HENP C++ physics libraries.
Go-HEP provides packages (in pure Go, no C++ required) to:

- read ROOT files (and their content: TH1x, TH2x, TTrees, TGraph)
- read/write YODA and numpy files
- read/write HepMC files
- read/write SLHA files
- read/write Les Houches files
- read/write SIO and LCIO files
- fill, save and load 1D and 2D histograms, scatters and profiles
- create PDF/LaTeX, PNG, JPEG, SVG and interactive plots

But Go-HEP also provides more physics-oriented packages:

- fmom: 4-vectors
- fads: a fast detector simulation toolkit, a reimplementation of C++ Delphes in (concurrent) Go,
- fastjet: a reimplementation of C++ FastJet, in concurrent Go,
- fit: a MINUIT-like minimization package,
- heppdt: HEP particle data tables,
- pawgo: a Physics Analysis Workstation in Go (REPL, plots, I/O)

We will first re-introduce – quickly – the concurrent programming building blocks of Go, its great development environment and tools (easy refactoring, quick deployment, fast edit/compile/run cycle, etc…).
We then describe two packages (rootio and fads) enabling physics analyses.
Finally, the performances (CPU, VMem) of two applications built with Go-HEP will be compared to their C++ counterparts: Delphes and Rivet.

Primary author: BINET, Sebastien (IN2P3/LPC)

Presenter: BINET, Sebastien (IN2P3/LPC)

Session Classification: Track 3: Computations in Theoretical Physics: Techniques and Methods

Track Classification: Track 3: Computations in Theoretical Physics: Techniques and Methods