JuliaHEP Notes

This document is public

Next Meeting: ?
Google Meets Link: (https://meet.google.com/xnq-xawz-fec)

We will follow the Julia Community Standards https://julialang.org/community/standards

We're following the trail set by JuliaHPC

Introduce yourself here (Name, institution, url/twitter/whatever)
- Adam Lyon, Fermilab (Muon g-2/Quantum Computing), homepagehttps://lyon-fnal.github.io, github
- Jerry Ling, Harvard (ATLAS), github
- Misha Mikhasenko <mikhail.mikhasenko@cern.ch>, CERN (LHCb), github
- Tamas Gal, ECAP (KM3NeT), https://github.com/tamasgal
- Kamil Ziemian, (currently don't working in any experiment)
- Johannes Schumann <johannes.schumann@fau.de>, ECAP (KM3NeT), https://github.com/8me
- Jan Strube, (Belle II, ILC) https://github.com/jstrube
- Jim Simone, Fermilab (Lattice QCD) github
- Alexei Strelchenko, Fermilab (Lattice QCD, PPS) https://github.com/alexstrel/QJulia
- John Hearns, Dell HPC Engineer, ex ALEPH (yea I am old) Dell HPC
- Jean-Loup Tastet, NBI Copenhagen (pheno, SHiP, ATLAS STA), github.com/JLTastet
- Oliver Schulz <oschulz@mpp.mpg.de>, MPI for Physics (GERDA, LEGEND), github.com/oschulz
- Simeon Schaub, bachelor student at KIT (bachelor thesis with CMS) https://github.com/simeonschaub
- Ivan Rojas(independent researcher) github/RG:ivan_rojas7

Julia HEP Related Talks and Papers
- Performance of Julia for High Energy Physics Analyses

Collection of topics for the next meeting
- UnROOT.jl updates

2021-06-09 Meeting 3-4pm (US Central Time) [Your Timezone]

Google Meeting URL **https://meet.google.com/xnq-xawz-fec**

Agenda (feel free to add and edit):
- Would people be interested in hearing about ttH event classification based on CMS data using Julia? - Simeon -- Sure, sounds good [Adam]
  - GitHub Repo: https://github.com/kit-cn-cms/LorentzLayerNetworks.jl
- I would like to get your opinion on distribution fitting: functions and pdfs with parameters (AlgebraPDF.jl) [Misha]
- Also, some thoughts on `data analysis for HEP in Julia` - my proposal for JuliaCon lightning talk got accepted
- Tamas Gal will give Julia introduction talk (with a bit of astroparticle physics in mind) at the ESCAPE Summer School (Dat Science for Astronomers, Astroparticle & Particle Physics): https://indico.in2p3.fr/event/20306/
  Here is the time slot: https://indico.in2p3.fr/event/20306/sessions/14896/#20210616

2021-01-27 Wednesday 9am-10am (America/Chicago) Meeting

See here for your timezone - we may need to go to 10:30 to fit everything in
~~Google Meet URL: https://meet.google.com/jaa-wnif-gjn~~

Agenda (feel free to edit):
- New introductions [5 min]
- Recent successes with Jula in HEP? [5 min]
- Intro to CxxWrap projects (Jan Strube) [15 min]
  - Overview of how to set up projects
  - Example: LCIO.jl
  - Discussion of the status of Minuit2.jl
- Discussion of fitting [30 min]
  - Jump.jl - Tamas [10 min]
  - Bat.jl - Oliver [20 min]
  - Discussion [Remaining time]
- Let's talk about GPUs and OneAPI at the Feb meeting

2020-12-11 Meeting

- Action items:
  - Adam will look into getting someone from Distributions.jl to attend a meeting in the near future so we can talk about our use cases
  - Alexei - want to give us a talk about OneAPI.jl ? Remember, we aren't GPU experts (at least Adam isn't)
  - Jan - want to give us a talk about making binaries?
  - Tamas - want to give us a talk about Jump.jl?

Agenda: [Adam's notes are interspersed]
- *Introductions*
    - Adam - HPC for analysis with HDF5 & MPI
    - Alexei - Lattice QCD, GPUs (NVidia & Intel)
    - Jan - Writing LCIO.jl (bindings to LCIO)
    - Jerry - Reweighting matrix elements
    - Johannes - Using Julia for neutrino analysis
    - John - CERN computing group
    - Kamil - Interested in Julia
    - Mikhail - Phenomenology - Julia for angular distributions / complex analysis
    - Tamas - Writing UnROOT.jl - Root I/O in pure Julia. Wrote Corpuscles.jl

- *Short perspective on current development of CERN Root?*
  Can no longer interface directly to Root since ROOT.jl and CXX.jl are no longer maintained. Root's use of LLVM conflicts with Julia's and ROOT.jl had a special julia executable that worked around that. For regular ntuples or histograms without custom objects, using UpROOT.jl or UnROOT.jl may be sufficient.

- *Successes with Julia in HEP?*
    - Adam - Early success running Julia on Cori with 1,280 processes spread over 20 nodes for Muon g-2 histogram production (one histogram for now).
    - Jan - Using Julia with students for complex geometry. A student did a Convex Hull algorithm for 3D cluster of particles
    - Mikhail - so far Julia wass great for small / individual projects
    - Alexei - Using Julia to try out GPUs. Compare with highly optimized LQCD codes. Looking at OneAPI toolkit (there's a Julia package now). Important to improve inter-node communication latencies. Lattice QCD is interested in Julia for developing I/O models. Perhaps Alexei can give us a little talk about OneAPI.

- *What needs work?*
    - *Analysis front (.root I/O, statistical tools)*
      Need fitting tools - like RooFit and RooStats. Interfacing to Python is a inconvenient (hard to deal with virtual environments). Note that many students are using Root in lab classes, e.g. to read oscilloscope data, especially for the i/o. This makes using Julia for analysis in labs difficult, though Up/UnROOT may be a solution.

    - *Upstream (particle generator?)*
      There are a lot of upstream dependencies (C++, Fortran). Jan has built binaries and put them in Yggdrasil. Perhaps Jan can give us a talk about making binaries.

    - *Tooling for smooth migration (PyCall works great, Cxx is broken)*

[I think we covered this elsewhere]

○ *Using CxxWrap.jl to work with C++, instead of Cxx.jl*
Cxx.jl is not the way forward. This means we need to ship a .jll with libraries. See upstream topic above

○ *Distributions: fitting and generation*
Distributions.jl is very active. But there's a lot missing that we need. Can't fit for all parameters. Mixed models don't work. Need a package like Roofit. Mikhail has a separate prototype package (AlgebraPDF.jl). Perhaps we can invite someone from Distributions.jl and talk about our use cases. Does Distributions.jl work on GPUs? We think so.

○ *HEP specific plotting recipes*
Tamas uses PGFPlotsX.jl directly (without Plots.jl). Plots.jl is nice, but plots break when backends change (so stability is a problem - also hard to deal with backends with different capabilities). Makie is interesting, but needs a GPU (so hard to run on a GPU-less remote resource).

○ *What is better IJulia or Pluto.jl? Pedagogical question*
IJulia is difficult, because setup scripts for experiments break the python environment. Jupyter is also hard to setup. Pluto facilitates running on multiple nodes easily [Adam's note: I don't think this is a problem for IJulia]

○ *Minimization: errors, stopping criteria.*
Optim.jl is like Minuit (BFGS is the same algorithm as Minuit - but maybe modern - Minuit uses fixed depth instead of auto-depth). Also, Munuit has an advanced stopping criteria calculationg EDM (). The problem is that Minuit is its own ecosystem (returns uncertainties, contours) Nothing like this in Julia (errors are now addressed in AlgebraPDF.jl). Julia has basic functionality, but needs more. If we were to wrap Minuit, would calling the Julia function evaluation be performant [Adam's note - if compiled, I think yes]. Tamas uses Jump.jl (apparently it can do minimization). Perhaps Tamas can give us a little talk about how he uses Jump.jl

● *Plan for future meetings?*
We all agreed to meet in the future! We will communicate on the Julia Slack #physics channel and Adam will open a new Discourse question. The plan is Adam will send out a Doodle Poll soon after the new year and we'll aim for a meeting in mid-January. We'll try to not have the meeting always at the same day/time to avoid some people being locked out due to their schedules. But realize it may be impossible to accommodate everyone.