

Minutes for the 15th April 2020

HSF Frameworks WG Meeting ([link](#))

Attendees: Chris Jones, Daniel Sherman Riley, Graeme Stewart, Hadrien Benjamin Grasland, Martin Ritter, Charles Leggett, Thomas Kuhr, Benedikt Haegner, Kyle Knoepfel, TJ Khoo, Attila Krasznahorkay

Introduction

- Brief excursion from multithreading topics to discuss Belle II's framework.
- This WG has about 1 hour to meet at the HSF/WLCG virtual meeting in May.
- Proposal is to (1) devote the first half to organizing code for smaller/less complex frameworks to take advantage of modern architecture, and (2) devote the second half to discuss possible multithreading topics.
- A specific plan will be discussed and suggested in the days ahead.

Belle II Framework Efforts

- Belle II is a B-factory, producing asymmetric e^+/e^- collisions on the Upsilon(4S) peak.
- Roughly 10 tracks per event.
- Framework written from scratch, supporting C++ computation, Python configuration, and support for ROOT, Geant4, etc.
- Parallelization is at the event-level, using shared POSIX memory and communicating across multiple processes
- Processes are forked as late as possible to maximize memory sharing
- Belle II expects that they cannot “afford” a multithreading migration, due to limited skills and efforts of developers/users
- Online and offline frameworks are the same modulo their configurations.
- Encourage use of Python as much as possible when doing analysis.
 - Jupyter meant mainly as a prototyping platform, the “final” configurations are then run as “regular python scripts”.
- Infrastructure and organization in place to ensure quality of code (CI, code shifts, memory checks, etc.)
- Built for multiple versions of RHEL and Ubuntu OS-es
- Simulation typically goes in “one step” from event generation to the reconstruction output
 - The whole processing takes 2-3 seconds/event. Doesn't make sense to write out intermediate formats.
- Basf2 fairly inter-connected with the rest of the Belle II software
 - Would not be easy to extract just the “framework” by itself at this point