DUNE framework LDRD

https://indico.fnal.gov/category/1415/attachments/154813/201467/FY2022_LDRD_Description_Knoepfel.pdf

Kyle J. Knoepfel (*in absentia*)

LDRD report @ SCD Projects Meeting

18 August 2022

Since last time (June 16)

July 21 update canceled due to Snowmass

- **SIST summer internship is over**
  - Tyler Terwilliger explored a graph-based framework approach (using OneTBB's flow graph).
  - No earth-shattering contributions toward furthering the LDRD, but it helped us learn under what conditions a flow-graph approach could be successful.
- **Began analysis of current DUNE workflows**
  - DUNE's hope is that their eventual framework will make it easier to manage memory well.
  - To that end, I profiled memory usage and CPU efficiency for various DUNE *art* jobs to get a baseline.
  - Some memory issues could be addressed by improving LArSoft algorithms (used widely by DUNE).
  - I've implemented *some* improvements to LArSoft (pull requests are forthcoming). But fully addressing this problem is off-scope and will require dedicated effort.
- **Technical progress**
  - I've met with Chris Jones and Marc Paterno to discuss the data model and the programming model.
  - I'm exploring an API where users register functions with the framework similar to how C++ functions are bound to Python in pybind11 (https://pybind11.readthedocs.io/en/stable/basics.html#creating-bindings-for-a-simple-function)
  - This approach provides a clean separation between framework and user code.