EIC Detector-1 Software Decision

Topic: Reconstruction Framework

Point of Contact: Wouter Deconinck

Discussion Date(s): June 29, 2022

Meeting Link: https://indico.bnl.gov/event/15644/

Endorsed by:

Wouter Deconinck (University of Manitoba)
Shujie Li (LBL)
Cristiano Fanelli (W&M)
Torre Wenaus (BNL)
John Lajoie (ISU)
Markus Diefenthaler (Jefferson Lab)
Joe Osborn (BNL)
Christopher Dilks (Duke)
David Lawrence (JLab)

Overview:

The reconstruction framework at the EIC aims to take advantage of heterogeneous computing and evolution of available hardware to transform collected and simulated data into reconstructed quantities. Over the lifetime of the reconstruction framework, it should allow for modularity to swap in and out individual reconstruction algorithms. The role of the reconstruction framework must include event reconstruction starting from simulated/collected events until reconstructed tracks and clusters, but may also include simulation at the front-end and may continue through physics reconstruction on the back-end (kinematics, energy flow, etc).

Requirements:

- The reconstruction framework must be able to run on both simulated events and real data.[1] Even if there may be algorithms that use truth information (or even require truth information, initially), the reconstruction framework itself should allow for running without truth information.

[1] "We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure." Statement of Principles, 2a.

- The reconstruction framework must be able to take advantage of heterogeneous computing resources (multiple cores, GPUs, etc).[2]
- The reconstruction framework must encourage modular approaches to algorithm development, using defined interface layers.[3]
- Algorithms must be implemented using the selected data model, and ensure that data (event data, geometry description, and algorithm parameters) are kept separate from the algorithm itself.
- Algorithms must be implemented in the framework independently from any scheduling strategies; an algorithm must not need to know how it is orchestrated, whether it is running in parallel, in single or multithreaded mode, concurrent or not, in online or offline analysis mode.
- The reconstruction framework must be open source, accessible to the entire community, and managed by a sustainable core team.[4]
- The reconstruction framework must be able to pass (and add) metadata and so-called slow control information to the output it produces, so input files are not needed and output files can stand on their own.
- The reconstruction framework must be able to run in streaming readout mode, that is:
    - with access to only parts of an event (single detector, single sector),
    - with events (or parts of events) appearing out of sequence,
    - individual algorithms must not rely on an algorithm-specific internal state to be able to make sense of disconnected parts of events.

Additional assessment criteria

- Amount of ‘boilerplate’ code that must be written by algorithm developers.
- Ability by the framework to avoid e.g. memory errors through interface enforcement mechanisms (e.g. const passing).
- Ability for shared algorithm development between the two EIC detector collaborations (and/or outside of the EIC).
- Use of modern and sustainable coding practices, including in the code written by algorithm developers and other contributors.
- Demonstration of performance in production environments.

Options: (TBC)

- Jana2: https://jeffersonlab.github.io/JANA2/
- Gaudi: https://gaudi.web.cern.ch/gaudi/

[2] “We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.” Statement of Principles, 3a.
[3] “We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.” Statement of Principles, 4c.
[4] “We will embrace our community.” Statement of Principles, 7.

Presenters: (TBC)

- Jana2: David Lawrence
- Gaudi: Sylvester Joosten

Discussion:

The live notes capture the details of the discussion, in particular through questions and answers provided there. The POC gives a personal summary of the discussion in the paragraphs that follow.

In the discussion, it became clear that both proposals are technically able to meet the requirements, but that in both cases there are uncertainties as well.

In the case of Gaudi, it was pointed out that the documentation is very out of date, that the CERN-based developers are not responsive to our concerns, and that this imposes a risk on the continued reliance on Gaudi for the future. Gaudi does not include streaming readout, which is provided in the case of LHCb by the Allen package.

As for Jana2, there is a large number of algorithms and services that will have to be developed in a short time before October. In recognition of this, Jefferson Lab is committing 1 FTE per year for the continued development, support, and training of the collaboration in jana2, and is mindful of the need to concentrate additional FTEs in the short term to manage a translation of existing algorithms from Gaudi.

The structure of algorithms in Gaudi and jana2 is similar, which will make exchange, reuse or translation of algorithms between two choices much less effort-intensive than it took to write the algorithms originally. Nevertheless, a few services are of the highest priority: the input/output layer based on podio (already in progress), and the interface with the DD4hep geometry.

Convener Summary:

The working group conveners recommend JANA2 as the reconstruction framework.

Although both Gaudi and JANA2 are technically able to meet the requirements, there is too much risk in depending on Gaudi which is not focused on a community outside of LHCb. The efforts invested in the already written juggler algorithms will be able to be reused with relatively minor effort in JANA2 algorithms. Likewise, the efforts invested in fun4all algorithms may be able to be reused.

The translation of the relevant Gaudi/juggler and fun4all algorithms will be completed by the Jefferson Lab EPSCI group by October 2022, aided by the 1 FTE-year per year committed by Jefferson Lab to the support of JANA2 for EIC.

The working group conveners point out that continued engagement with the key4HEP project through the development of modular reconstruction algorithms and functional programming approaches is desirable.

The Jefferson Lab EPSCI group is encouraged to develop JANA2 into a community project where developers from outside Jefferson Lab are valued at all stages of software development.