Date: Dec. 29 2017

EIC Detector R&D Progress Report

Project ID: eRD20
Project Name: Developing Simulation and Analysis Tools for the EIC
Period Reported: Jul. 14 2017 - Dec. 29 2017
Project Leaders: M. Diefenthaler, A. Kiselev
Contact Persons: M. Diefenthaler (mdiefent@jlab.org), A. Kiselev (ayk@bnl.gov)

Project members
ANL Whitney Armstrong, Sergei Chekanov, David Blyth
BNL Elke-Caroline Aschenauer, Alexander Kiselev, Jerome Lauret, Christopher Pinkenburg
Fermilab Stefan Prestel
INFN Trieste Andrea Bressan
Jefferson Lab Markus Diefenthaler, David Lawrence, Dmitry Romanov, Maurizio Ungaro
SLAC Makoto Asai, Andrea Dotti
William & Mary Wouter Deconinck

Abstract

The EIC will revolutionize our understanding of the inner structure of nucleons and nuclei. Developing the physics program for the EIC, and designing the detectors needed to realize it, requires a plethora of software tools and multifaceted analysis efforts. Many of these tools have yet to be developed or need to be expanded and tuned for the physics reach of the EIC. Currently, various groups use disparate sets of software tools to achieve the same or similar analysis tasks such as Monte Carlo event generation, detector simulations, track reconstruction, event visualization, and data storage to name a few examples. With a long-range goal of the successful execution of the EIC scientific program in mind, it is clear that early investment in the development of well-defined interfaces for communicating, sharing, and collaborating, will facilitate a timely completion of not just the planning and design of an EIC but ultimate delivery of the physics capable with an EIC. The EIC Software Consortium aims to develop simulation and analysis tools for the EIC and to facilitate communication and collaboration among current and future developers and users.

Past

What was planned for this period?

In FY18, we are placing emphasis on:

- ❑ Monte Carlo Development
- ❑ Development of Detector Simulations with a focus on
  - ❑ "Geant4 Simulations"
  - ❑ "Interfaces and integration"
- ❑ Initiatives for:
  - ❑ "High-Performance Computing"
  - ❑ "Machine Learning"
  - ❑ "Web-Based User Interfaces"

We will continue our involvement in the "Future Trends in Nuclear Physics Computing" workshop and will continue building connections to the HEP Computing community via the HEP Software Foundation and the HEP Center for Computing Excellence. We have also started a new project on using containers to distribute the EIC software.

What was achieved?

Monte Carlo Development

E.C. Aschenauer, M. Diefenthaler, and S. Prestel organize with the mcNET Community a Monte Carlo workshop: "The 8th International Conference on Physics Opportunities at an ElecTron-Ion-Collider" (POETIC-8)", which is is an ideal opportunity to discuss Monte Carlo Event Generators (MCEGs) for upcoming ep and eA colliders, such as the EIC, LHeC and VHEeP facilities. A satellite workshop "MCEGs for future ep and eA facilities" will be held on March 22-23 2018 at University of Regensburg, Germany, to review the MCEGs for ep and eA processes and discuss the requirements and developments for the MCEGs for the science program at the EIC, LHeC, and VHEeP. The workshop is organized by scientists working on the EIC, E.C. Aschenauer (BNL) and M. Diefenthaler (JLab), and scientists from the mcNET community, S. Plätzer (Vienna) and S. Prestel (FNAL)."

At the end of August 2018, E.C. Aschenauer and A. Bressan had a productive meeting with Prof. H. Spiesberger in Mainz. H. Spiesberger is the author of HERACLES and the leading expert on radiative corrections. They have established a collaboration with H. Spiesberger on our project for a radiative correction library. Few weeks after the meeting, they received version of HERACLES stripped from all external dependencies so that it was possible to start including the radiative correction code in Pythia. The activity to include HERACLES in Pythia6, started soon after and is now well advanced, even if not fully finished and tested. The same approach cannot be followed for Pythia8 and we are presently studying possible alternatives.

Development of Detector Simulations

Geant4 Simulations Makoto Asai and Andrea Dotti are serving as the point of contact between the EIC Software Consortium and the Geant4 International Collaboration, representing the EIC community needs in the Geant4 collaboration, monitoring the progress, and making sure feedbacks are delivered to ESC in timely manner. One of the most important aspects for the success of the Geant4 simulation of EIC detectors is the correctness of the physics simulation modeling. While the efforts that have been done for the simulation of LHC detectors are a good starting point for the EIC simulation, it is however expected that tuning of the physics models will be needed to fully address the energy range of the EIC physics and peculiarities of EIC simulations. Together with Chris Pinkenburg, Makoto and Andrea are working on a physics list optimized and tuned for the EIC needs. The choice of physics models and transition regions between them require validation against published data and against alternative MC codes. Detector components for future EIC detectors are currently being tested at Fermilab's Test Beam facility. These tests cover the relevant energy ranges for the EIC and will be used to develop and verify the physics lists used in the accompanying Geant4 simulations. These test beam setups have been incorporated into the Geant4 simulations and comparisons are being published. The test beam data are analyzed within the same framework as the Geant4 based simulation which will result in a fast turnaround time when testing even the newest Geant4 beta versions. The ESC containers being currently developed will provide an easy access for Geant4 developers to the test beam results.

Common Geometry Interface After finalizing the document with the requirements for a geometry interface and making it available to the community (see section **Publications**), we have further investigated the necessary developments to achieve the common interface. Starting with the assumption that GDML is the standard to exchange geometry information, we have developed a plan to create a stand-alone lightweight MC hits library. After a review with the community at ESC meeting in FY17, we have decided to focus our efforts on creating a prototype interface for tracker style hits (i.e. position and energy deposition collections), to be stored in ROOT-format NTuples files. To guarantee the success of the project and an efficient use of resources, calorimeter style hits (i.e. accumulation of energy in cells) will be postponed and evaluated after the implementation of tracker hits. For a similar reason, the native reader of hits file from Numpy/Scipy is postponed: considering that it is always possible to read a ROOT file into a python program the python interface is not seen as a high priority at this stage.

Common IO Further progress has been made in the development of shared IO formats. A greater consensus was reached earlier on adopting EicMC, a feature-rich, flexible, self-descriptive data format based on Google protocol buffers. Substantial efforts have progressed to take the concept of a self-descriptive shared format beyond the MC level with the goal of facilitating development of data model interfaces throughout the entire process of event simulation and reconstruction. A new effort called ProIO has come about aiming to achieve this goal in a language-neutral way. In particular, language neutrality here means consistent access to shared IO in C++, Python, Java, and Go. Future work needs to determine how to best utilize the EicMC and ProIO tools at the MC level as well as in simulation and reconstruction. Priority is given to consolidation of the EicMC and ProIO in a single development, which would make best use of the concepts developed so far .

Unified track reconstruction Based on our feasibility study, we have decided to develop modular tracking based on the tracking software developed at ANL, BNL, and Jefferson Lab. We have started to define the interfaces for the track finding and fitting software (reconstructed hits, material geometry, magnetic fields) and are defining the milestones for our common development. Our modular tracking software will be developed as open source software so it can be easily contributed to by other nuclear physics projects. Several of the proponents of this proposal will facilitate adoption and real-world testing of the tracking software at upcoming experiments at BNL and Jefferson Lab. We will reach out to Paolo Calafiura (LBL) who has led the LHC Tracking Machine Learning Challenge to take advantage of the recent HEP developments.

Containers The ESC has identified container technology (Docker, Singularity, etc.) as a tool for distributing full EIC software stacks for users. Identical containers can be used on any platform (Linux, Windows, Mac OS X) and will carry all of the software needed to run an EIC simulation. Site-specific containers will be provided that bundle their own software with ESC non-site-specific software. The ESC has generated a document providing guidelines for the structure of the containers (see section **Publications**). The common structure will allow users to transfer familiarity with one site's container to another's while investigating and comparing physics topics between multiple sites. Containers are expected to be published for general consumption early 2018. The second phase will soon follow with containers customized as applications appropriate for GRID, cloud or HPC use.

The EIC Detector Advisory Committee recommended that we *"take a more active role in working with the detector consortia to help with the simulations and set up a process to easily implement new detector configurations to optimize the detector design."* We have followed up on the recommendation by starting a new project on containers and reviewing the software that has been developed for the EIC. The containers with our projects on a common geometry interface, a common IO, and an unified tracking will certainly allow us to follow directly the committee's recommendation.

The steering committee of the EIC User Group (EICUG) has asked us to summarize the status and plans for the EIC software at the recent EICUG Detector Discussion meeting at Temple University. Our review covered the whole palette of existing EIC-related GEANT-based software frameworks (GEMC from JLab, fun4all and EicRoot from Brookhaven, Argonne software suite) as well as the legacy fast smearing generator (eic-smear code developed at Brookhaven), EIC R&D PID Consortium software, existing tools to model EIC Interaction Regions and few advanced standalone simulation applications. Reflecting the Software Consortium philosophy it was stated that in the present diverse environment and given a very limited dedicated manpower one should not expect various software efforts to merge with each other but we should rather focus on developing common interfaces and shared tools, which in particular would allow one to exchange the detector geometries between different frameworks and to design the software algorithms, libraries and tools in a way they can later be used in various environments rather than be strictly bound to a particular framework and/or toolkit. It was also anticipated that in these early days of EIC the users should not expect to find a "perfect EIC simulation environment" under any of the existing frameworks, but rather be ready to contribute to the common development by either providing feedback to the software maintainers or, even better, by contributing to the code base directly.

What was not achieved, why not, and what will be done to correct?

The deliverable of a web-based event display depended on the full funding of our proposal and had to be dropped from the list of FY18 deliverables. Apart from that, we expect to accomplish all of our FY18 goals in the remaining nine months of the fiscal year.

Future

What is planned for the next funding cycle and beyond? How, if at all, is this planning different from the original plan?

We have not yet made significant progress with our initiatives on high-performance computing and machine learning.

High-Performance Computing

We would like to take full advantage of advances in scientific computing, in particular the advance of Exascale Computing and start to collaborate with the three ASCR-operated computational science user facilities: the Argonne Leadership Computing Facility at ANL, the National Energy Research Scientific Computing Center (NERSC) at LBNL, and the Oak Ridge Leadership Computing Facility at ORNL. In FY18, we will prepare EIC-related HPC projects and will begin a dialogue with the three ASCR-operated computational science user facilities.

Machine Learning

Over the last several years neural networks are getting a new spin. Better understanding of basic principles, emerging new methods and faster hardware (GPUs, FPGAs, TPUs) pushed the evolution of deep neural networks (DNN) so that machine learning has made a quantum leap, bringing speech recognition, better image processing, self driving cars and much more. Nuclear and high energy physics may benefit from this resurgence too, because most of the new approaches may be transferable to both online and offline physical analysis problems. E.g., convolutional and fully convolutional neural networks (CNN, FCN) are efficient and directly applicable to pattern recognition, PID, clustering and parameter extraction of cell based detectors like calorimeters, RICH, cell vertexes and others. Three-dimensional convolutional networks are efficient for volumetric or spatiotemporal pattern recognition which can be applied to PID, DAQ and track finding and fitting. Multimodal deep learning can be used for data fusion - combining outputs from

multiple detectors. DNN may also be used for anomaly detection, providing additional possibilities for data selection. All of the above examples of Neural Networks integrate well with classical analysis methods and with each other. Thus, DNN can be employed in a leading or supplementary role in a large number of tasks.
This proposal is about creating an EIC-centric machine learning framework. Instead of trying to be general, the development of the framework will be solving narrow, EIC specific, priority ordered problems, utilizing DNN and other modern machine learning methods and tools. We will work with Keras, a machine learning framework with a well-designed API that uses TensorFlow, CNTK, and Theano backends for defining abstract, general-purpose computation graphs. Keras allows the building of deep learning models by clipping together high-level building blocks and running them on backends, which in turn utilize CPUs, GPUs, or even new specially designed Tensor Processing Units (TPUs) in the most efficient way. In FY18, we will document selected examples for using DNN in NP and work on a roadmap for an EIC-centric machine learning framework.

What are critical issues?

There are no critical issues.

Additional information:

None.

Manpower

The eRD20 members work on a best effort basis on our projects. The eRD20 funds are presently not used to pay any positions, including the PostDocs on our project.

In FY17, eRD20 funds were used to support an undergraduate student project at Jefferson Lab: Sam Markelon (UConn) worked with M. Diefenthaler and M. Ungaro on web-based user interfaces for EIC software. His work was connected to the eRD20 objective of planning for the future with forward compatibility. UConn provided a stipend for S. Markelon; we paid for his accommodation at Jefferson Lab.

In FY18, eRD20 funds will also be used to support undergraduate student projects, to the extent possible.

External Funding

No external funding has been obtained.

Publications

We are documenting our work for the community and have made two documents available:

- Geometry Description and Detector Interface: The document lists the requirements for a common geometry and detector interface.
- ESC Containers: The document lists the guidelines for containers.