HEP reconstruction at HPC centers: an approach for efficient utilization of resources

S. Berkman, G. Cerati (*), J. Kowalkowski (*), M. Paterno, S. Sehrish (FNAL), T. Peterka, O. Yildiz (ANL)

(*) email contact: cerati_AT_fnal.gov, jbk_AT_fnal.gov

Introduction

In the last two decades, data processing for High Energy Physics (HEP) experiments has mostly relied on the Open Science Grid (OSG) as well as dedicated High Throughput Computing (HTC) data centers. Typically workflows exploit multi-core architectures by submitting multiple processes per node or, in the best cases, they support multi-threading at the event level which allows for a reduction in the overall memory usage. An important limitation of this approach is that, since applications access nodes with a variety of hardware, they cannot be optimized for a specific architecture; in other words, advanced features (such as SIMD processing units, many cores, low latency data exchange between nodes) cannot be exploited, and configurations need to tailor to a “lowest common denominator” among the available architectures.

Increasing investments in the construction of High Performance Computing (HPC) centers dedicated to science applications have recently enabled the exploration of a different paradigm, where HPC centers complement the traditional HTC workflows and boost resources for sample production campaigns. Most efforts so far have relied on container technology [1] (e.g. singularity, docker, etc.) as a solution to deploy the software stack of the experiments at HPC centers; in particular, this solution is needed since the distributed file system (cvmfs) that stores the compiled releases is typically not accessible from the HPC nodes. Containers are an effective solution to be able to run HEP workflows on HPC systems, but the workflows are not well optimized since they typically contain the standard software stack build, without any advanced instruction set and without including platform and vendor-specific optimized libraries.

The next generation of HEP experiments will produce a data volume orders of magnitude larger than the current ones, so data reconstruction will demand increased processing performance. For this reason, it is mandatory to explore solutions that will allow a more efficient utilization of HPC resources for HEP workflows. It is very important that the solutions explored also allow for a transparent inclusion of Artificial Intelligence (AI) applications, where usage within HEP is becoming more and more popular. HPC systems are being constructed to efficiently support AI workloads at large scale.

In this letter, we present plans for the development of a first workflow for HEP reconstruction which targets an efficient utilization of unique resources available at HPC facilities. This includes not only the many-core processors and multi-node infrastructure, but also high-bandwidth distributed memory and object store features for accelerating I/O. The processing of the data from the ICARUS experiment [2,3] has been chosen as a test case. An important goal is that, even in its first version, this workflow is not just a demonstration but can be part of a central sample production campaign for the experiment. Under these conditions, the collaboration will directly benefit from the performance enhancements provided by HPC resources.

Optimization of algorithms

Traditionally HEP reconstruction algorithms have been designed and implemented as serial applications. Therefore, most of the reconstruction workflows cannot be efficiently executed in highly parallel environments, such as HPC centers. However, HEP data offer several opportunities for parallelization, so algorithms can be re-designed in such a way that they can exploit them. This is particularly true for data from liquid argon time projection chambers (LArTPC), such as ICARUS and DUNE [4]. These detectors are modular in nature, so the data can be processed independently for each modular unit, as well as for each event. Instances of modular units are cryostats, TPCs, planes, and wires.

Examples of algorithms that, in the last few years, have been re-designed in order to exploit data and instruction parallelism are numerous, demonstrating that this is a viable solution to achieve significant speedups for reconstruction. Among these, it has been demonstrated that speedups can be achieved for LArTPC reconstruction

algorithms and specifically for the hit finder algorithm that is a significant contributor to the ICARUS reconstruction workflow [5]. Results show that in order to obtain significant vectorization speedups, it is necessary to compile the code using a custom configuration (i.e. *icc* and AVX-512 for the hit finder). This may include a dedicated compiler and core support libraries separate from the ones that may be used by the rest of the code.

Therefore, in order to fully exploit the resources available on each node of HPC centers, it is of high importance that the available setup allows for the customization of compiler options for specific algorithms. For this purpose, we are testing Spack [6] as the build system for the ICARUS software stack. Spack is a flexible package manager supporting multiple versions, configurations, platforms, and compilers that is well-suited and specifically designed for use on supercomputers. Early tests are promising, and demonstrate that the software stack can be successfully built with Spack.

Workflow definition

An important part of our plan is to use existing and experimental ICARUS and LArSoft [7] algorithms and configurations along with current multi-threaded features of the art framework [8] to manage event processing within one compute node. The art framework application and I/O modules will be enhanced with ASCR[1]-provided workflow libraries, distributed memory services, and data-parallel programming tools. These new features will permit users to run from their laptop or workstation command line as they do today, but also have the option of running across nodes in local clusters using the MPI launch facility, or though multi-node batch submission systems available at NERSC, ALCF, and Fermilab. These additional modes will permit scaling to thousands of nodes at HPC facilities with a straightforward configuration. Algorithm sequences within the framework configuration will no longer be confined to one node thanks to the distributed memory I/O services. This feature will permit node allocations that differ by algorithm and depend on the load characteristics of that algorithm or special accelerator needs. Multi-node algorithms and processing within one event will also be permitted, but that will not be an initial aim for algorithm enhancements. Our workflow targets facility-wide access to physics objects within events of large-scale datasets at any node allocation size. This will allow automatic load balancing and work partitioning features to be added over the course of this project and it also eliminates the need for file merging and concatenation steps during processing.

Our ICARUS workflow will initially target running of the signal processing, hit finding, clustering, and track finding stages of production processing but this may change depending on the needs of the experiment. We will start from artdaq raw data event fragments and end with selected data converted back to ROOT for furthering processing by the experiment.

Plan of work

Substantial preparatory work for the project has already been completed with the help of our HPC partners from DOE's ASCR Program [9] SciDAC Institutes [10]. In particular, the core software tools needed to move forward with this project have been identified and tested. These include Spack for building art and LArSoft, and HDF5 for highly parallel storage and access to large datasets. We have demonstrated success for NOvA analysis workflows using ASCR-developed DIY [11] for data-parallel programming across nodes. HEPnOS [12], a distributed memory service designed specifically for high-performance access to HEP physics objects on HPC systems, is also being tested at large scale for neutrino event selection procedures. We have successfully demonstrated mapping LArSoft waveform and analysis data and NOvA CAFana data from ROOT to HDF5. Many of these tools are also used for event generator workflow running at HPC center at large scale [13].

[1] Advanced Scientific Computing Research

Work is ongoing towards assembling the different parts into a first version of the workflow, to be run on the Theta supercomputer at the Argonne Leadership Computing Facility. In particular, the first build of the ICARUS software stack at Theta using Spack is ongoing, as well as the implementation of data conversion procedures from the ROOT-based format used in the input and output file to the transient distributed memory object store (HEPnOS) and the HDF5 format used in the workflow. Next steps will be the customization of the spack build for specific packages and the test of different workflow parameters for an optimal usage of the resources. A detailed validation of the output to assess the compatibility with the standard production workflow will be performed.

The first tests may quickly evolve and include more features in the HPC workflow. In particular, we plan to include AI-based steps in the chain, some of which may be written in Python, and study the interplay of traditional and AI algorithms in the optimization of the workflow.

Conclusions

This letter details the status and the plans for a workflow to run HEP reconstruction jobs at HPC centers, with the goal of an enhanced utilization of advanced features of HPC centers. The initial target experiment and HPC facility are ICARUS and Theta at ALCF, respectively. The first application of the project is expected to be completed by the end of the Snowmass2021 program, and can be used to inform the community for planning ahead towards a more efficient usage of HPC resources for HEP data processing.

References

[1] https://www.nersc.gov/research-and-development/user-defined-images/ and https://www.alcf.anl.gov/support-center/theta/singularity-theta

[2] ICARUS Collaboration, Design, construction and tests of the ICARUS T600 detector, Nucl. Instrum. Meth. A 527, 329 (2004).

[3] M. Antonello, et al., A Proposal for a Three Detector Short-Baseline Neutrino Oscillation Program in the Fermilab Booster Neutrino Beam, arXiv:1503.01520 [physics.ins-det] (2015).

[4] DUNE Collaboration, The DUNE Far Detector Interim Design Report Volume 1: Physics, Technology and Strategies, arXiv:1807.10334 [physics.ins-det] (2018).

[5] S. Berkman, et al., Reconstruction for Liquid Argon TPC Neutrino Detectors Using Parallel Architectures, 24th International Conference on Computing in High Energy and Nuclear Physics, arXiv:2002.06291 (2020).

[6] https://spack.io/

[7] E. L. Snider and G. Petrillo, LArSoft: toolkit for simulation, reconstruction and analysis of liquid argon TPC neutrino detectors, J. Phys. Conf. Ser. 898, no. 4, 042057 (2017)

[8] C. Green, et al., The art framework, J. Phys.: Conf. Ser. 396 022020 (2012)

[9] https://www.energy.gov/science/ascr/advanced-scientific-computing-research

[10] https://rapids.lbl.gov/

[11] https://www.mcs.anl.gov/~tpeterka/software.html#diy

[12] https://xgitlab.cels.anl.gov/sds/hep/HEPnOS

[13] https://github.com/HEPonHPC/pythia8-diy