Portable Parallelization Strategies - a CCE project

Oliver Gutsche for the HEP-CCE project

The hardware landscape is evolving. Traditional multi-/many-core CPU architectures best suited for distributed high-throughput computing (dHTC) are now being augmented by architectures utilizing compute accelerators (GPUs, FPGAs, etc.). The next generation of High-Performance Computing (HPC) facilities will deliver most of their compute power via accelerated sub-systems. This development is also not standing still, with new installations using newer and more innovative architectures coming online approximately every 5 years. The computational power available at HPC centers, in terms of raw installed Flops, is many orders of magnitude more than that available to HEP experiments through their pledged dHTC resources.

Effectively using these HPC systems, however, is a challenge for the code bases of HEP experiments that can consist of many million lines of C++ code. The code base consists of algorithms that encode scientific innovation and are mostly written by physicists who are not experts in writing highly specialized code for accelerators. It is also not feasible to re-write these algorithms when new HPC installations come online with different accelerator architectures, because of the long development cycles and stringent validation requirements.

The HEP Center for Computational Excellence (HEP-CCE)[1] is a U.S. Department of Energy (DOE) cross-cutting initiative to promote HPC utilization within the particle physics community including data-intensive applications, scientific simulations, and data movement and storage. An important part of the center's activities is to enhance connections with DOE's Advanced Scientific Computing Research (ASCR) program[2]. The center is investigating the challenges and possibilities of using next-generation computing architectures, especially accelerators, for HEP data-intensive applications.

The "Portable Parallelization Strategies" (PPS) project is part of the center's activities with the goal to investigate solutions that allow HEP code to run relatively well on many architectures, for example traditional CPU and accelerator-based architectures. Such a solution would allow the community to design algorithms and write code once that then can be executed on the various architectures without major changes.

To achieve this goal, the HEP-CCE project will investigate a range of portable programming solutions (such as Kokkos/RAJA, SYCL/DPC++, and OpenMP/OpenACC) on three cases from the ATLAS, CMS and DUNE experiments:

- FastCaloSim, fast calorimeter MC simulations
- Patatrack pixel tracking

[1] https://hepcce.org
[2] ASCR/HEP Exascale Requirements Review Report: https://arxiv.org/abs/1603.09303

- Wire-Cell Toolkit, detector response simulations

All three use cases have CUDA implementations (of varying degree) to test the feasibility of GPU acceleration, and present different algorithms and code complexities in the HEP software ecosystem

The HEP-CCE project has defined a suite of metrics to evaluate the suitability of each portability solution based on three categories:

- **Productivity**: ease of porting/learning, code impact/change, build time, debugging, etc.
- **Portability**: architectural portability, temporal portability (how sustainable the model is...)
- **Performance**: absolute performance (sustained/peak) while it is understood that portability is more important than ideal performance

The goal is to make recommendations to the broader HEP community by the end of 2022 through reports, publications and community outreach.

Currently, the CCE-PPS project is working on the Kokkos implementation of the use cases.

A portable programming model that allows the average physicist to contribute algorithmic code easily while allowing for reasonably performant execution on a variety of architectures will be essential for the future for HEP applications. We therefore ask for consideration of this important topic in the Snowmass 2021 process.