Develop of Surrogate Models for Analysis Code: proposal for DOE NP DE-FOA-0002490

David Lawrence, Malachi Schram

March 2021

Abstract

This proposes to develop tools and techniques to produce AI/ML surrogate models to replace expensive algorithms in reconstruction/analysis software. This will allow transitioning existing codes to run on heterogeneous hardware. The expectation is that this will lead to improved processing speed and promote development of codes with improved correctness in the future.

1 Introduction

Algorithms currently used in reconstruction/analysis codes are written in high-level languages by system experts. The more complex algorithms may take significant time to develop and debug for correct and efficient operation. HPC facilities used to run these programs at scale have been moving to more heterogeneous hardware designs, partly to accommodate more AI/ML based algorithms. To utilize these heterogeneous resources, existing codes either need to be re-written or replaced with AI/ML based algorithms. An ML model developed to replicate the function of a piece of software written in a high-level language is called a *surrogate*. This proposal is to develop techniques for producing surrogates from existing codes so that the models can be run on heterogeneous hardware without the need of rewriting the code for that specific platform. We propose two development tracks for this. One for generic tools to profile and replace expense function calls with ML surrogates. The other is to replace larger complex algorithms not contained in a single function call with a ML surrogate.

The benefits of this work would include:

- Enable existing codes to use heterogeneous hardware without significant manpower investment
- Provide flexibility to run existing codes on multiple types of hardware by producing hardware agnostic models
- Improve correctness and reduce development time for future algorithms since authors will not need to simultaneously optimize both.

2 Inserting ML-generated Surrogates into Inner-application Pipelines to Leverage Heterogeneous Hardware

The goal of this project is to develop a machine learning pipeline that replaces computationally expensive function calls with ML-generated surrogates (MLGS). We will produce a framework that supports the end-to-end process that identifies costly functions calls and build/calibrate/validate MLGSs on heterogeneous computing architectures (CPU, GPU, FPGA, etc.). This project would provide the NP community a framework to leverage upcoming hardware architecture and reduce the computational time.

3 Surrogate Model Development for Complex Algorithms in Multi-threaded Call Stacks

The most expensive parts of the CLAS12 and GlueX analysis codes are those responsible for charged particle track reconstruction (*tracking*). These are complex pieces of software spread over numerous function calls with dependencies on calibration constants, geometry maps, and run conditions. The software has significant complexity in both its inputs and its outputs that developing a surrogate model will require expert input and an understanding of the software in some detail. Efficient use of external heterogeneous resources may also require bundling of tasks from multiple execution threads leading to additional synchronization points in the code. This sub-project would focus on developing a surrogate for one of these tracking codes. It would have an immediate impact on the ongoing experimental program. In addition, the exercise would inform how to address the problem in other, similarly complex algorithms.

4 Resources

The project would require some resources (*TBD*)

References