Performance-portability of the Albany multi-physics finite element code on the road to exascale

Jerry Watkins[1], Max Carlson[1,2], Kyle Shan[3], **Irina Tezaur**[1]

[1]Sandia National Laboratories, [2]University of Utah, [3]Micron Technology

As HPC architectures become more heterogeneous, scientific software must adapt to take advantage of potential performance capabilities. This talk focuses on attaining performance-portability of the Sandia Albany multi-physics C++ finite element code [1]. In an effort to avoid architecture specific programming, we have adopted within the Albany code base a mixed (MPI+X) approach of utilizing high level abstractions and the Kokkos programming model [2] for performance portable code across a variety of different architectures. By utilizing Kokkos and various packages within Trilinos, a significant amount of progress has been made towards developing a framework that supports high resolution simulations on pre-exascale systems and beyond. We will present some key performance-portability developments for the land ice modeling application implemented within Albany and known as ALI, covering topics ranging from finite element assembly, to linear solvers, to automated performance analysis/tuning. We will demonstrate that the same code runs correctly and efficiently, with reasonable scalability, across a variety of computer architectures including the most current generation of GPUs. We will also provide some perspectives on evaluating, improving and maintaining performance-portability of a production code such as Albany.

[1] A. Salinger, *et al.* "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *Int. J. Multiscale Comput. Engng* 14(4) (2016) 415-438.

[2] C. Edwards, C. Trott, D. Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. Journal of Parallel and Distributed Computing, 74(12):3202–3216, 2014.