Barrier-Free Parallel–Adaptive Scheme for Asynchronous Spacetime Discontinuous Galerkin Methods

Robert Haber[1], Amit Madhukar[1], Reza Abedi[2], Volodymyr Kindratenko[3]

[1]Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, 2101A Mechanical Engineering Lab, Urbana, IL, 61801
[2]Mechanical, Aerospace & Biomedical Engineering, University of Tennessee Knoxville (UTK) / Space Institute (UTSI), 411 B. H. Goethert Parkway, Tullahoma, TN 37388
[3] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 3050E NCSA, 1205 W. Clark St., Urbana, IL 61801, USA

This presentation describes a new architecture for parallel–adaptive simulation using asynchronous Spacetime Discontinuous Galerkin (aSDG) methods. The new architecture abandons two mainstays of conventional parallel finite element computation: the bulk synchronous parallel (BSP) model and the Domain Decomposition Method (DDM). The BSP model is naturally incompatible with the aSDG method and is also emerging as an obstacle to achieving exascale computing goals. Therefore, we opt for a barrier-free asynchronous design. In previous work, we obtained near-perfect scaling with static domain decompositions and non-adaptive aSDG models using Charm++, but we found that revising the domain decomposition for load balancing was too cumbersome and far too slow to keep up with the fine granularity of aSDG adaptive meshing. Our solution was to parallelize the fundamental aSDG operations — spacetime patch construction, local finite element solution, and adaptive remeshing — at their common granularity: a single patch. In the case of adaptive meshing, this requires a modified algorithm called lazy refinement. In view of the uniform locality of the resulting aSDG algorithm, the advantages of domain decomposition into contiguous subdomains vanish. Load balance is controlled by the assignments of vertices in the space-like front mesh to the available cores, as these assignments determine where spacetime patches are handled. Rebalancing the load reduces to modifying the per-vertex pointers that determine these assignments, a simple, lightweight procedure that is executed at the same patch-level granularity as the rest of the algorithm. We present results from strong scaling studies to demonstrate the effectiveness of this new parallel architecture.