Parallel PDE-Based Simulations Using the Common Component Architecture

Lois Curfman McInnes\textsuperscript{1}, Benjamin A. Allan\textsuperscript{2}, Robert Armstrong\textsuperscript{2}, Steven J. Benson\textsuperscript{1}, David E. Bernholdt\textsuperscript{3}, Tamara L. Dahlgren\textsuperscript{4}, Lori Freitag Diachin\textsuperscript{4}, Manojkumar Krishnan\textsuperscript{5}, James A. Kohl\textsuperscript{3}, J. Walter Larson\textsuperscript{1}, Sophia Lefantzi\textsuperscript{6}, Jarek Nieplocha\textsuperscript{5}, Boyana Norris\textsuperscript{1}, Steven G. Parker\textsuperscript{7}, Jaideep Ray\textsuperscript{8}, and Shujia Zhou\textsuperscript{9}

\textsuperscript{1} Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
[mcinnes, benson, larson, norris]@mcs.anl.gov
\textsuperscript{2} Scalable Computing R & D, Sandia National Laboratories (SNL), Livermore, CA, USA
[baallan, rob]@ca.sandia.gov
\textsuperscript{3} Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA
[bernholdt, kohlja]@ornl.gov
\textsuperscript{4} Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA
[dahlgren, diachin2]@llnl.gov
\textsuperscript{5} Computational Sciences and Mathematics, Pacific Northwest National Laboratory, Richland, WA, USA
[manojkumar.krishnan, jarek.nieplocha]@pnl.gov
\textsuperscript{6} Reacting Flow Research, SNL, Livermore, CA, USA
slefant@ca.sandia.gov
\textsuperscript{7} Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA
sparker@cs.utah.edu
\textsuperscript{8} Advanced Software R & D, SNL, Livermore, CA, USA
jairay@ca.sandia.gov
\textsuperscript{9} Northrop Grumman Corporation, Information Technology Sector, Chantilly, VA, USA
szhou@pop900.gsfc.nasa.gov

Summary. The complexity of parallel PDE-based simulations continues to increase as multimodel, multiphysics, and multi-institutional projects become widespread. A goal of component-based software engineering in such large-scale simulations is to help manage this complexity by enabling better interoperability among various codes that have been independently developed by different groups. The Common Component Architecture (CCA) Forum is defining a component architecture specification to address the challenges of high-performance scientific computing. In addition, several execution frameworks, supporting infrastructure, and general-purpose components are being developed. Furthermore, this group is collaborating with others in the high-performance computing community to design suites of domain-specific component interface specifications and underlying implementations.
This chapter discusses recent work on leveraging these CCA efforts in parallel PDE-based simulations involving accelerator design, climate modeling, combustion, and accidental fires and explosions. We explain how component technology helps to address the different challenges posed by each of these applications, and we highlight how component interfaces built on existing parallel toolkits facilitate the reuse of software for parallel mesh manipulation, discretization, linear algebra, integration, optimization, and parallel data redistribution. We also present performance data to demonstrate the suitability of this approach, and we discuss strategies for applying component technologies to both new and existing applications.

10.1 Introduction

The complexity of parallel simulations based on partial differential equations (PDEs) continues to increase as multimodel, multiphysics, multidisciplinary, and multi-institutional projects are becoming widespread. Coupling models and different types of science increases the complexity of the simulation codes. Collaboration across disciplines and institutions, while increasingly necessary, introduces new social intricacies into the software development process, such as different programming styles and different ways of thinking about problems. Added to these challenges, the software must cope with the multilevel memory hierarchies common to modern parallel computers where there may be three to five levels of data locality.

These challenges make it clear that the high-performance scientific computing community needs an approach to software development for parallel PDEs that facilitates managing such complexity while maintaining scalable and efficient parallel performance. Rather than being overwhelmed by the tedious details of parallel computing, computational scientists must be able to focus on the particular part of a simulation that is of primary interest to them (e.g., the physics of combustion) and employ well-tested and optimized code developed by experts in other facets of a simulation (e.g., parallel linear algebra and visualization). Traditional approaches, such as the widespread use of software libraries, have historically been valuable, but these approaches are being severely strained by this new complexity.

One goal of component-based software engineering (CBSE) is to enable interoperability among software modules that have been developed independently by different groups. CBSE treats applications as assemblies of software components that interact with each other only through well-defined interfaces within a particular execution environment, or framework. Components are a logical means of encapsulating knowledge from one scientific domain for use by those in others, thereby facilitating multidisciplinary interactions. The complexity of a given simulation is decomposed into bite-sized components that one or a few investigators can develop independently, thus enabling the collaboration of scores of researchers in the development of a single simulation. The glue that binds the components together is a set of common, agreed-upon interfaces. Multiple component implementations conforming to the same external interface standard should be interoperable, while providing flexibility to accommodate different aspects such as algorithms, performance characteristics, and coding styles. At the same time, the use of common interfaces facilitates the reuse