Environmental Modelling & Software 78 (2016) 134–149

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

Managing complexity in simulations of land surface and near-surface processes

Ethan T. Coon [a, *], J. David Moulton [b], Scott L. Painter [c]

[a] Computational Earth Sciences, Los Alamos National Laboratory, USA
[b] Applied Mathematics and Plasma Physics, Los Alamos National Laboraotry, USA
[c] Climate Change Science Institute, Environmental Sciences Division, Oak Ridge National Laboratory, USA

ARTICLE INFO

Article history:
Received 7 August 2014
Received in revised form
16 December 2015
Accepted 29 December 2015
Available online xxx

Keywords:
Multiphysics
Framework
Directed acyclic graph
Land surface modeling
Thermal hydrology

ABSTRACT

Increasing computing power and the growing role of simulation in Earth systems science have led to an increase in the number and complexity of processes in modern simulators. We present a multiphysics framework that specifies interfaces for coupled processes and automates weak and strong coupling strategies to manage this complexity. Process management is enabled by viewing the system of equations as a tree, where individual equations are associated with leaf nodes and coupling strategies with internal nodes. A dynamically generated dependency graph connects a variable to its dependencies, streamlining and automating model evaluation, easing model development, and ensuring models are modular and flexible. Additionally, the dependency graph is used to ensure that data requirements are consistent between all processes in a given simulation. Here we discuss the design and implementation of these concepts within the *Arcos* framework, and demonstrate their use for verification testing and hypothesis evaluation in numerical experiments.

© 2016 Elsevier Ltd. All rights reserved.

Software Availability

Name of software: Arctic terrestrial simulator (ATS)
Developers: Ethan Coon
Contact: ecoon_at_lanl.gov
Year First Available: 2014
Hardware Required: Flexible, laptops to supercomputers
Software Required: Linux/OSX, Amanzi and its dependencies (see following entry for Amanzi)
Software Availability: Source and documentation available at https://software.lanl.gov/ats
Cost: Free, Open Source (three-clause BSD license)
Name of Software: Amanzi
Developers: Multi-institution team led by J. D. Moulton, see https://software.lanl.gov/ascem/amanzi for more information.
Contact: moulton_at_lanl.gov
Year First Available: 2014
Hardware Required: Flexible, laptops to supercomputers
Software Required: Linux/OSX with C++/C/Fortran compilers, Message Passing Interface (MPI), CMake tools. A bootstrap configuration tool is provided that automatically downloads and builds additional third party libraries (TPLs).
Software Availability: Source and documentation is available at https://software.lanl.gov/ascem/amanzi
Cost: Free, Open Source (three-clause BSD license)

1. Introduction

Increasing computational power has enabled not only the increase of spatial and temporal resolution of simulations in the Earth sciences, but also an increase in complexity. More physical processes are included, with a higher degree of physical fidelity in each process representation. This growth in complexity represents a fundamental shift for multiphysics simulators from the old paradigm, in which a small number of process models are coupled together in a limited number of pre-defined combinations, to a new paradigm, in which many, varying processes are coupled in nearly arbitrary permutations. In moving from a few to many process models, identifying which processes are critical to simulate a given

* Corresponding author.
E-mail address: ecoon@lanl.gov (E.T. Coon).

http://dx.doi.org/10.1016/j.envsoft.2015.12.017
1364-8152/© 2016 Elsevier Ltd. All rights reserved.

phenomena is often a fundamental part of the science, requiring flexible coupling strategies and eliminating a statically configured coupling strategy as an option.

The requirements of two complex environmental simulation codes with significant subsurface focus, the *Arctic Terrestrial Simulator* (*ATS*) and *Amanzi*, exemplify the challenges inherent in emerging Earth Science applications. The need to simulate the hydrologic response of Arctic tundra to warming temperatures, and the fate of carbon stored in Arctic soils have motivated development of *ATS*. *ATS* performs coupled simulations of soil thermal hydrology with freeze/thaw cycles, topological evolution due to melting subsurface ice, surface flows with freezing/thawing, snow processes, surface energy balance, soil biogeochemistry, and vegetation processes (Painter et al., 2013). This is a computationally challenging application because of the number and complexity of the controlling physical processes and constitutive models, very strong nonlinearities that necessitate implicit coupling, the need for complex unstructured meshes requiring advanced discretizations, and different process representations on different subdomains ranging from three-dimensional (subsurface processes), to two-dimensional (surface processes), to one-dimensional (snow models), to zero-dimension (surface energy balance and carbon decomposition models). This application stresses the need for a flexible and dynamic multiphysics framework. There is significant uncertainty in the physical phenomena necessary to accurately predict carbon emissions, and error introduced by coupling strategies has not yet been quantified. Developing software explicitly configured for all of the possible combinations for processes and process couplings would be extremely difficult and error prone. Instead, there is a strong need for approaches that allow for a flexible, hierarchically organized, and dynamically configured model.

Similar considerations apply to the Environmental Management program within the Department of Energy, which oversees the remediation and closure of DOE sites storing legacy waste from the development of nuclear weapons and related technologies. Although, at a high-level these sites have flow and reactive transport in common, there are signficiant differences in the heterogeneous subsurface environment, complex biogeochemsitry, and external forcing that affects contaminant transport. In addition, a variety of engineered systems such as tanks, flow barriers, and injection wells are often used and can be modeled with varying levels of abstraction and fidelity. Thus, to underpin scientifically defensible decisions and strategies for cleanup and remediation, risk and performance assessments use a graded and iterative approach. This approach first establishes simplified models and then iteratively enhances geometric and process level complexity to identify and characterize the key processes and assumptions that are needed to efficiently reach a defensible decision. This need led to the development of the Advanced Simulation Capability for Environmental Management (ASCEM) program, and its simulator *Amanzi*. Moreover, it required that *Amanzi* support a flexible and extensible design that would support run time selection of process models, and interoperability with established biogeochemical reaction capabilities (Moulton et al., 2012). An important feature of this design is that it significantly improves the ability of developers to meet software quality assurance requirements because it enables testing components both in isolation and in integrated settings.

The requirements for *ATS* and *Amanzi* and the broader shift toward greater process complexity in land surface and subsurface simulations represent both a challenge and an opportunity for basic research in advanced computational scientific software. Although in traditional multiphysics efforts, data, data dependencies, and model coupling strategies were managed by an "omniscient coder,", new paradigm applications require a more abstract view. Specifically, pursuing this new level of abstraction creates the opportunity for a well-designed programming model to gather information about the physical system, informing automated methods for exposing concurrency within the application, thus helping to bridge the productivity gap between application software and extreme-scale machines. Additionally, flexibility gained by following abstract interfaces and frameworks significantly improves the efficiency of developing and running new simulations to test scientific hypotheses, advance understanding and make predictions.

In fact, the increasing interest in process-rich simulations with ever higher-fidelity is a common challenge across disciplines that have embraced simulation. Thus, strategies for managing complexity in process-rich simulations are beginning to appear in various applications and include data managers (Slattery et al., 2013; Larson et al., 2005), dependency graphs (PLASMA, 2014; Berzins et al., 2012; Kale et al., 2007), and process couplers (Craig et al., 2012; Redler et al., 2010; Peckham et al., 2013; Dunlap et al., 2013). Data managers are more common in simulations considering multiple domains, such as atmosphere-ocean-land climate models, or in engineering applications with solid and fluid domains. Tools such as the Data Transfer Kit (Slattery et al., 2013), the Model Coupling Toolkit (Larson et al., 2005), and OASIS (Redler et al., 2010) focus on mapping fields between differing meshes, in parallel, at domain interfaces. However, as most codes deal with a single discretization assumed from the start of development, these typically assume that fields are statically located at a single mesh entity.

Dependency graphs are seeing more and more attention as a potential tool for extreme-scale computing. These graphs have been used, for instance, to manage fine-grained concurrency in dense linear algebra calculations in PLASMA (Plasma, 2014) and coarse-grained concurrency in finite-element simulations in Phalanx (Notz et al., 2012). Similarly, task graphs, which form dependency graphs for evaluation processes, have been used to exploit concurrency in task-based parallelism strategies. This approach has been used in several codes and programming models, including Uintah (Berzins et al., 2012), Charm++ (Kale et al., 2007), and Legion (Bauer et al., 2012).

Generalized process couplers, which provide interfaces for components to implement and thereby be coupled to other components, are a common strategy in the Earth sciences, and have been approached in an ad hoc way in many codes. However, several efforts deserve note for their generality for complex problems. In particular, climate codes often must couple modules from ocean, atmosphere, sea ice, and land processes into a common framework. Efforts toward this problem include the CESM flux coupler CPL7 (Craig et al., 2012), the Earth System Modeling Framework (Hill et al., 2004), and the Flexible Model System (Balaji, 2005). CSDMS (Peckham et al., 2013) uses similar approaches for surface processes. In these efforts, interfaces for coupling are required of each component, and once a component implements that interface it may be plugged into every other system component. Typically, these are limited to sequential coupling. Recently, feature modeling, a form of meta-analysis about these types of couplers, has been performed on several couplers in the literature by Dunlap et al., 2013. These types of efforts result in clean, modular code, albeit at a relatively coarse granularity generally focused on coupling through fluxes.

Although these efforts have contributed capability and experience that are valuable for geoenvironmental modeling, none alone are sufficient for meeting the challenges presented by emerging simulation needs in ecohydrological (e.g. ATS) and environmental management applications (e.g. Amanzi). In this work, we present a

Download English Version:

https://daneshyari.com/en/article/6962596

Download Persian Version:

https://daneshyari.com/article/6962596

Daneshyari.com