Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Future Generation Computer Systems 21 (2005) 878–895
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

Novel runtime systems support for adaptive compositional modeling in PSEs

Srinidhi Varadarajan, Naren Ramakrishnan *

Department of Computer Science, Virginia Tech., Blacksburg, VA 24061, USA

Received 24 September 2003; received in revised form 30 October 2003
Available online 8 February 2004

Abstract

Complex problem solving environments (PSEs) rely on runtime systems support for adaptive composition and control of scientific computations. This paper presents a PSE runtime support solution through a novel combination of two technologies—*Weaves*, a source language independent parallel compositional framework that operates through reverse-analysis of compiled object files, and *runtime recommender systems* that aid in dynamic knowledge-based application composition. Domain-specific adaptivity is realized by recommendation of code modules, supported by a sophisticated checkpointing framework for runtime control. A core set of "adaptivity schemas" are provided as templates for adaptive composition of large-scale scientific computations. Implementation issues, motivating application contexts, and preliminary results are described.

Keywords: Problem solving environments; Checkpointing; Recommender systems; Application composition systems; Adaptivity

1. Introduction

Problem solving environments (PSEs) are evolving from standalone systems to complex, networked, entities seamlessly integrating geographically disparate resources in a single application. Specific trends contributing toward this evolution are the increasing componentization of scientific codes, emerging system infrastructures such as the Grid [11], and the runtime constraints posed by novel computational science applications. To be effective in these emerging environments, PSEs must provide high-level, powerful, computational primitives within the context of the emerging system infrastructures. This requires both an understanding of the architectural assumptions of today's computational systems and an appreciation for how disciplinary scientists do computational science. The focus of this paper is runtime systems support that is cognizant of the operational issues underlying PSE infrastructure and is flexible enough to accommodate diverse application scenarios.

An area where runtime systems support holds great promise is in the engineering of adaptivity—adaptivity

* Corresponding author. Tel.: +1 540 231 8451; fax: +1 540 231 6075.
E-mail addresses: srinidhi@cs.vt.edu (S. Varadarajan), naren@cs.vt.edu (N. Ramakrishnan).

0167-739X/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2003.12.020

in terms of algorithm selection, architectural tuning, and exploiting the underlying scientific usage contexts [11]. We posit a broad picture of adaptivity here, one which is not restricted to identifying partitioning parameters, modifying data decompositions, or parallel scheduling; instead, adaptivity is proposed at a more logical unit of algorithms and object codes. This viewpoint leads to scientific codes being organized in a model-based framework for adaptive composition, execution, and performance analysis [1]. We use the term *compositional modeling* in this paper to collectively refer to all three aspects of *model specification* (how to define the composed elements?), *model execution* (how to execute composed codes?), and *model analysis* (how to use performance information from execution to evaluate and improve models?).

These considerations lead us to identify two important requirements for runtime systems support in PSEs. First, runtime systems support should enable a transparent transition path for composing and executing legacy codes, without requiring that they be rewritten to achieve this functionality. Second, runtime adaptivity should allow the dynamic selection, reconfiguration, and execution of code modules, taking into account performance considerations, problem characteristics, and dynamic system infrastructures.

1.1. Solution approach

Our solution approach for runtime systems support is two-pronged: (i) a novel compositional system with checkpointing support for deployment over HPC platforms, and (ii) realizing domain-specific adaptivity through a runtime recommender system. A core set of "adaptivity schemas" constitute a reconfigurable approach to steering and managing large-scale scientific computations.

In this context, a high-level problem specification (e.g., "solve this elliptic PDE with a relative accuracy of 10^{-6} and time less than 600 s") is provided to a recommender system that makes an initial recommendation of code modules (e.g., "use a finite-difference discretizer with red-black ordering"). These code modules are communicated to the compositional system as a "configuration", which are then scheduled and executed; as the computation progresses (e.g., the PDE gets discretized and the resulting linear system appears to be ill-conditioned), feedback is provided to the runtime recommender through the checkpointing mechanism, which uses this information to perhaps dynamically insert a preconditioner before the linear solver in the solution loop. The configuration is updated with this selection, and the computation is re-scheduled. This interplay between the compositional system (which supports object-based composition, migration, and checkpointing) and the runtime recommender (which enables dynamic selection of code modules) leads to a novel runtime framework for scientific computations.

1.2. In this paper

Section 2 identifies two core computational technologies that form the basis of our solution for runtime systems support. Section 3 elaborates on how these technologies are integrated to provide novel systems support for PSEs. Section 4 identifies a set of "adaptivity schemas" that can be used as templates for realizing many complex, adaptive, scientific computations. Section 5 presents early results and outlines work in progress. A concluding discussion placing this work in context of emerging trends is provided in Section 6.

2. Core computational technologies

Our approach to supporting adaptive compositional modeling centers on two core computational technologies: the *Weaves* parallel compositional framework, and data-driven runtime recommender systems. We discuss them in detail in the context of a real scientific application.

2.1. Motivating application

Our driver application involves the idea of *collaborating partial differential equation* (*PDE*) *solvers* [9] for solving heterogeneous multi-physics problems. For instance, simulating a gas turbine requires combining models for heat flows (throughout the engine), stresses (in the moving parts), fluid flows (for gases in the combustor), and combustion (in the engine cylinder). Each of these models can be described by an ODE/PDE with various formulations for the geometry, operator, and boundary conditions. The basic idea here is to replace the original multi-physics problem by a set of smaller simulation problems (on simple geometries) that need