Lecture Notes in Computer Science 4340

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK
Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
University of Surrey, Guildford, UK
Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
Friedemann Mattern
ETH Zurich, Switzerland
John C. Mitchell
Stanford University, CA, USA
Moni Naor
Weizmann Institute of Science, Rehovot, Israel
Oscar Nierstrasz
University of Bern, Switzerland
C. Pandu Rangan
Indian Institute of Technology, Madras, India
Bernhard Steffen
University of Dortmund, Germany
Madhu Sudan
Massachusetts Institute of Technology, MA, USA
Demetri Terzopoulos
University of California, Los Angeles, CA, USA
Doug Tygar
University of California, Berkeley, CA, USA
Moshe Y. Vardi
Rice University, Houston, TX, USA
Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Radu Prodan Thomas Fahringer

Grid Computing

Experiment Management, Tool Integration, and Scientific Workflows

Springer

Authors

Radu Prodan
Thomas Fahringer
University of Innsbruck, Institute for Computer Science
Technikerstr. 21a, 6020 Innsbruck, Austria
E-mail: {radu,thomas.fahringer}@dps.uibk.ac.at

Library of Congress Control Number: 2006939015

CR Subject Classification (1998): C.2, D.4, F.3, H.4, H.3, C.4, I.2.8

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-69261-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-69261-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Markus Richter, Heidelberg
Printed on acid-free paper SPIN: 11965800 06/3142 5 4 3 2 1 0

To our families

Preface

In the last decade, the interest in computational Grids has increasingly grown in the scientific community as a means of enabling the application developers to aggregate resources scattered around the globe for solving large-scale scientific problems. As applications get larger, more complex and dynamic, the use of software tools becomes vital for tuning application parameters, identifying performance leaks, or detecting program defects. Extensive efforts within academia and industry over the last decade resulted in a large collection of tools for practical application engineering. Available tools of broad interest include program source and structure browsers, editors, static program analysers, performance predictors, optimising compilers, schedulers, execution control and monitoring environments, sequential and parallel debuggers (providing deadlock detection and deterministic replay mechanisms), checkpointers, data and execution visualisers, performance analysers, or various program tracers.

Despite all these extensive efforts, building applications that can effectively utilise the Grid still remains an art due to the lack of appropriate high-level tools to support the developers. In this monograph, we address four critical software development aspects for the engineering and execution of applications on parallel and Grid architectures.

First of all, existing available performance analysis tools target single application execution which is not sufficient for effective performance tuning of parallel applications. The most popular performance metrics such as speedup or efficiency require repeated execution of the application for various machine sizes for which no automatic tool support exists so far. Additionally, parallelising and tuning of applications for a certain compute platform requires repeated experimentation for various data distributions, loop iteration scheduling strategies, or compiler optimisation options, which is known to be an NP-complete problem.

Second, tool portability is critical since tools are often based on a monolithic design that does not isolate the inherent platform dependencies required to support advanced and in-depth analysis. For example, when using a new parallel system the users must in most cases learn about and familiarise themselves with new tools with different functionality and interfaces, which is in many cases very time consuming and can be a major barrier in using novel modern computer architectures.

Third, existing tools cannot be used in cooperation on the same application instance to enhance the performance and correctness debugging engineering process since they are not designed for interoperability and often are based on incompatible instrumentation or monitoring systems.

Fourth, the workflow model that recently emerged as a new attractive paradigm for programming loosely coupled Grid infrastructures requires novel tools that offer appropriate high-level support, including abstract specification mechanisms, optimised scheduling, and scalable fault-tolerant execution, which are of paramount importance to effectively running distributed large-scale applications. These topics attract a lot of interest within the Grid community that aims to evolve the Grid to a commodity platform that transparently aggregates high-performance resources scattered around the globe in a single virtual supercomputer for performing scientific simulations.

In this monograph, we first propose a new directive-based language called ZEN for compact specification of wide value ranges of interest for arbitrary application parameters, including problem or machine sizes, array or loop distributions, software libraries, interconnection networks, or target execution machines. We design the ZEN directives as problem-independent with global or fine-grain scopes that do not change the semantics of the application, nor require any application modification or special preparation. Irrelevant or meaningless experiments can be filtered through well-defined constraints. Additionally, the ZEN directives can be used to request a wide range of performance metrics to be collected from the application for arbitrary code regions.

Based on the ZEN language, we develop a novel experiment management tool called ZENTURIO for automatic experiment management of large-scale performance and parameter studies on parallel and Grid architectures. ZENTURIO offers automatic analysis and visualisation support across multiple experiments based on the performance and output data collected and organised in a common shared data repository. In contrast to existing parameter study tools, ZENTURIO requires no special preparation of the application, nor does it restrict the parametrisation to input files or to global input arguments. We validated the functionality and usefulness of ZENTURIO on several real-world parallel applications from various domains, including theoretical chemistry, photonics, finances, and numerical mathematics.

We designed ZENTURIO as a comprehensive distributed service-oriented architecture for interoperable tool development based on the latest state-of-the-art Web and Grid services technologies. We illustrate how a service-oriented architecture facilitates the integration of a broad set of tools and enables a range of useful tool interoperability scenarios that facilitate the engineering effort of applications. We illustrate a variety of novel adaptations of state-of-the-art Web technologies for Grid computing which anticipated several existing standardisation efforts.

Based on the ZENTURIO experiment management architecture, we propose a generic optimisation framework that integrates general-purpose metaheuristics for solving NP-complete performance and parameter optimisation problems in an exponential search space specified using the ZEN experiment specification language. We illustrate a generic problem-independent realisation of the search engine using a genetic algorithm that allows new optimisation problems to be formulated through appropriate objective functions, for example, a performance metric using the ZEN language. We illustrate three case studies that instantiate the framework for Grid workflow scheduling, throughput scheduling of parameter studies, and performance tuning of parallel applications on the Grid using irregular array distributions.

Finally, we propose a timely approach for modelling and executing scientific workflows in dynamic and heterogeneous Grid environments. We introduce an abstract formal model for hierarchical representation of complex directed graph-based workflows using composite activities (such as parallel and sequential loops or conditional activities) interconnected through control and data flow dependencies comprising advanced collective communication patterns such as broadcast, scatter, and gather. We propose and comparatively analyse three heuristic-based algorithms for scheduling two real-world scientific workflows from material science and meteorology domains. The scheduled applications achieve good performance on the Austrian Grid environment using advanced runtime techniques such as partitioning, workflow optimisation, and load balancing. We design a steering algorithm that performs runtime monitoring and workflow schedule adaptations which ensure that certain quality of service performance contracts are preserved during execution of the workflow. We conclude with a classification of the most important performance overheads that may slow down the performance of scientific workflows and validate them through several experiments.

Innsbruck, October 2006

Radu Prodan
Thomas Fahringer

Contents

List of Figures

List of Tables

List of Algorithms