Introduction

Ioana Banicescu

Department of Computer Science and Engineering,
Mississippi State University,
P.O. Box 9637, 300 Butler Hall, Mississippi State, MS 39762, USA
E-mail: ioana@cse.msstate.edu

Thomas Rauber

Computer Science Department,
University Bayreuth,
95440 Bayreuth, Germany
E-mail: rauber@uni-bayreuth.de

Gudula Rünger

Computer Science Department,
Chemnitz University of Technology,
09107 Chemnitz, Germany
E-mail: ruenger@informatik.tu-chemnitz.de

Biographical notes: Ioana Banicescu is an Associate Professor in the Department of Computer Science and Engineering at Mississippi State University (MSU), and Chair of the Scientific Computing research focus group. She received her Diploma in Engineering (electronics and telecommunications) from the Polytechnic University, Bucharest, and her MS and PhD in Computer Science from the Polytechnic University, New York. She is the recipient of a number of awards for research and scholarship. Her research interests include parallel algorithms, scientific computing, scheduling theory, load balancing algorithms, performance analysis, optimisation, and prediction. She is a senior member of the IEEE, a member of ACM, SIAM and AMS, and presently serves on the Executive Board of the IEEE Technical Committee on Parallel Processing.

Thomas Rauber received the Diploma, PhD and Habilitation in Computer Science from University Saarbruecken, Germany in 1986, 1990, and 1996, respectively. He became a Professor in Computer Science Department at the Martin-Luther-University Halle-Wittenberg in 1996. Since 2002, he is Professor at the University of Bayreuth where he leads a research group on parallel and distributed systems. His research interests include: software development and programming environments for parallel and distributed systems; compiler and library support for parallel and distributed systems; program transformations for locality improvements; mixed task and data parallel models; parallel and distributed scientific computing.

Gudula Rünger received her Diploma and PhD in Mathematics from the University of Cologne, Germany, and her Habilitation in Computer Science from the University of the Saarland, Germany. Between 1997 and 2000 she was an Associate Professor in Computer Science at the University of Leipzig, and in 2000 she became a full Professor in the Computer Science Department at Chemnitz University of Technology, Germany. Her research interests include parallel and distributed programming, parallel languages and cost models, scientific computing, software tools and libraries for mixed programming models, and transformation approaches. Her multidisciplinary research includes collaborative projects with mathematicians, physicists, and engineers.

Although progress in parallel and distributed methodologies for scientific computing have been quite remarkable during the past years, this area of computer science remains still active, especially in topics concerning the relationship between performance and aspects such as: irregularity of applications and algorithms, adaptive characteristics of software and hardware, heterogeneity of hardware platforms, and flexibility of programming environments. Recent research activities include development of complex hardware architectures, including storage hierarchies or heterogeneous (parallel and distributed) computing platforms with large numbers of processors, as well as irregular applications that involve complex domain decomposition and hierarchical, adaptive and multi-level organisation of computation and data structures. The corresponding irregular algorithms comprise sparse, block-structured or adaptive data structures, and irregular, runtime-dependent computation and control structures.

Over time, to improve scientific applications' performance on sequential machines, several techniques in hardware and algorithm design, such as storage hierarchies and hierarchical domain decomposition, have been introduced. However, the simulation of large irregular problems still requires the use of parallel and distributed environments. The irregular and dynamically changing runtime behavior makes an efficient parallel realisation difficult, since the memory access patterns and the evolution of dynamic structures cannot be determined priori, and therefore, cannot be planned statically. Consequently, an efficient parallel implementation of this class of problems necessitates the exploitation of flexible programming environments as well as techniques to improve scalability.

This special issue is based on the Dagstuhl seminar *Adaptivity in Parallel Scientific Computing* which took place in May 2003 in the pleasant castle of Dagstuhl in Germany (www.dagstuhl.de). This seminar was a forum that brought together researchers working in different areas of parallel scientific computing and its applications, to solve scientific and industrially oriented problems. It provided a fertile environment for the participants to meet and exchange ideas, as well as to foster future research collaborations. Of particular interest was the exchange of experiences in interdisciplinary research projects. Topics covered by this seminar included:

- parallel numerical algorithms
- parallel implementation of irregular applications
- algorithms for memory hierarchies with enhanced locality of memory access
- libraries for supporting parallel scientific computing
- mixed task and data parallel executions on large parallel machines
- performance analysis evaluation and prediction
- compiler transformations for increasing the locality of memory references
- dynamic load balancing techniques
- partitioning and scheduling strategies
- heterogeneous computing (cluster and grid computing)
- combination of different programming models for heterogeneous parallel machines.

During the seminar, a number of presentations lead to formulation of interesting open questions followed by discussions on optimal integration of adaptivity at various levels of technology in application, algorithms and system development. The contributions in this special issue summarise some of these presentations and discussions to bring the topics to a broader audience. In the following paragraphs, we give a short summary of the contributions.

The paper 'Tinkertoy parallel programming: a case study with Zoltan' by K. Devine and B. Hendrickson advocates a toolkit approach for developing parallel applications. A toolkit provides a small set of related services that can be used as parts of many different applications. The use of Zoltan represents a specific attempt to use the toolkit approach for building parallel adaptive scientific applications.

A structured parallel programming approach for irregular applications with irregular domains, heterogeneous architectures, or non-dedicated architectures is promoted in the paper 'Irregularity handling via structured parallel programming' by M. Danelutto. In a structured parallel programming environment, the application programmer only needs to express the kind of parallelism to be exploited, while many details of the parallel implementation, including issues related to irregularity, are automatically taken care of by the programming environment.

The paper 'Supporting tasks with adaptive groups in data parallel programming' by J. O'Donnell proposes a technique for embedding task parallelism in a data parallel computation model. The use of an abstract SIMD machine allows a fast implementation of adaptable grouping and communication operations. This enables the application to adapt quickly to the demands of problems with irregular or dynamically changing structure.

The paper 'Foundations for the integration of scheduling techniques into compilers for parallel languages' by W. Zimmermann and W. Löwe shows how task-scheduling can be integrated into compilers for data parallel languages. A key technique is to consider tasks to be executed on more than one processor, known as malleable tasks. The introduction of malleable tasks allows the use of a hierarchical scheduling approach that takes into account function calls without the need for expansion.

Scheduling and load balancing on heterogeneous clusters are considered in 'Scalable loop self-scheduling schemes for heterogeneous clusters' by A.T. Chronopoulos et al. Distributed scheduling schemes suitable for parallel loops on heterogeneous computers are studied, and an extension is implemented in a hierarchical master-slave architecture.

Dynamic load balancing is addressed in the paper 'Performance evaluation of a dynamic load-balancing library for cluster computing' by R.L. Cariño, I. Banicescu, J.P. Pabico and M. Balasubramaniam. The new approach proposed herein combines dynamic loop scheduling strategies with resource management and task migration. Performance evaluations for a specific library of dynamic load balancing methods are presented.

The paper 'Speedup in solving differential equations on clusters of workstations' by D. Petcu proposes a performance model for iterative numerical methods on cluster architectures. The performance model is based on a data flow model and task graphs. Specific applications are recursive relations, parallel Runge-Kutta methods, and volume element methods for parabolic partial differential equations.

The paper 'Graphics hardware for scientific computation' by P. Lucas presents modern graphics processing units and their potential use for general purpose computations from parallel numerical algorithms. Hardware and support for possible programming languages are described.

The influence of memory hierarchies on irregular application programs is studied in 'A method to derive the cache performance of irregular applications on machines with direct mapped caches' by C. Scholtes. A probabilistic method to derive the cache performance of irregular applications based on the source code is presented. The method is applied to sparse matrix multiplication and sparse Cholesky factorisation.

In conclusion, the contributions in this special issue address many complex issues including application requirements for adaptivity in space and time, as well as requirements for improving the capacity to effectively use resources in heterogeneous environments. These topics span and integrate the work from many research areas: irregular scientific applications, adaptive algorithms, programming models and tools, problem solving environments for cluster and grid computing, and others. We believe that these contributions will inspire the readers to continue their research efforts towards an integrated view of adaptivity, allowing them in this way to make significant contributions to the advancement of science.

We thank all the participants of the seminar and the authors of the articles presented in this special issue for their work and fruitful discussions. We also thank the many referees for their detailed and constructive reviews that helped to improve the contributions. Finally we would like to thank the IBFI Schloss Dagstuhl for making this seminar on *Adaptivity in Parallel Scientific Computing* possible.