Available online at www.sciencedirect.com
SCIENCE DIRECT®
Parallel Computing 31 (2005) 649–652
PARALLEL COMPUTING
www.elsevier.com/locate/parco

Guest editorial

Heterogeneous computing

This special issue on heterogeneous computing is a follow-on of two well established workshops in the domain, namely HCW, the IEEE Heterogeneous Computing Workshop (held in Santa Fe in April 2004, in conjunction with IPDPS) and HeteroPar, the International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (held in Cork in July 2004, in conjunction with ISPDC).

Networks of computers are the most commonly available parallel architecture now. Unlike dedicated parallel computer systems, networks are inherently heterogeneous. They consist of diverse computers of different performances interconnected via mixed network equipments providing communication links of different speeds and bandwidths. Traditional parallel algorithms and tools are aimed at homogeneous multiprocessors and cannot be efficiently used for parallel computing on heterogeneous networks. New ideas, dedicated algorithms, and tools are needed to efficiently use this new type of parallel architectures.

This special issue gathers extended and revised versions of the best eight papers published in the above two workshops. We give a brief outline of the contents of each paper in the following paragraphs.

The first two papers deal with scheduling heuristics for heterogeneous systems. In the paper entitled ''A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems'', Tarek Hagras and Jan Janeček present a simple task-graph scheduling algorithm based on list-scheduling and task-duplication. They target execution on a bounded number of heterogeneous machines, linked by a general interconnection graph. Their analysis and experiments show that their approach can outperform higher complexity algorithms.

In the paper ''Mapping subtasks with multiple versions on an ad-hoc grid'', Sameer Shivle, Prasanna Sugavanam, H.J. Siegel et al. investigate various mapping and scheduling algorithms onto ad hoc grids. An ad hoc grid is a heterogeneous computing and communication system, all of whose components are mobile and have limited power capacity. A large application task composed of several communicating subtasks is to be mapped onto machines the ad hoc grid. All subtasks must be

0167-8191/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.parco.2005.04.001

executed, either in their full version, or in a degraded version which utilizes only a fraction of the resources that the full version requires, and produces only a fraction of the data output for the subsequent children subtasks. Thus, the degraded versions represent a reduced capability of lesser overall value, while consuming fewer resources. The goal is to assign resources so that the application meets an execution time constraint and the battery energy constraint, while minimizing the number of degraded versions used.

The next two papers are devoted to algorithmic issues. Parallel algorithms for traditional homogeneous distributed memory multiprocessors are designed to evenly load their processors. One process per processor is a typical configuration providing top performance for such parallel systems. There are two main approaches to parallel scientific programming for heterogeneous clusters. The first one is to modify the homogeneous parallel algorithm in order to load heterogeneous processors in proportion to their speed, still assuming the one-process-per-processor configuration of the parallel program. The other approach is to use the same homogeneous algorithm but run multiple processes per processor to try and balance the load of the processors this way. The paper ''Optimizing the configuration of a heterogeneous cluster with multiprocessing and execution-time estimation'' by Yoshinori Kishimoto and Shiuchi Ichikawa presents a case study applying the second approach to optimizing the execution of the High Performance Linpack benchmark on two heterogeneous clusters. The authors suggest performance models of execution of this application on a heterogeneous cluster and use the models to select the optimal subset of processors and determine the optimal number of processes running on each processor.

Javier Cuenca, Domingo Giménez and Juan-Pedro Martínez investigate the same approach in the paper ''Heuristics for work distribution of a homogeneous parallel dynamic programming scheme on heterogeneous systems''. The authors analyze how to adapt an application implementing a homogeneous parallel dynamic programming algorithm for efficient execution on a heterogeneous cluster. The application used in the paper solves the ''coin problem''. The authors suggest a performance model of the application in the form of function of the problem size, the parameters of the executing heterogeneous cluster and the parameters of the algorithm. It is assumed that the parameters of the model can be accurately estimated at runtime. Based on the estimation of the parameters, the application selects at runtime the optimal number of processes, the processors to use and the optimal mapping of the processes to the processors.

Instead of considering static scheduling and mapping techniques as in the previous papers, the next two papers address dynamic load balancing strategies. The paper ''Design and implementation of a novel dynamic load balancing library for cluster computing'' by Ioana Banicescu, Ricolindo L. Cariño, Jaderick P. Pabico and Mahadevan Balasubramaniam considers dynamic load balancing for applications with computationally intensive parallel loops. For such applications, dynamic load balancing allows to respond to variations in the system characteristics. Advanced dynamic load balancing algorithms are far from trivial to design and implement. Therefore, a library implementing the algorithms can be useful for

application developers. The paper presents such a library and its integration with a runtime system supporting object migration. The library provides a number of dynamic loop scheduling algorithms based on probabilistic analysis and has an open architecture.

There are two kinds of parallel systems: (i) capability based, aimed at minimizing the completion time of one big job, and (ii) capacity based, aimed at maximizing the number of completions of small jobs within a given time. Heterogeneous computing systems are an attractive platform for both capability and capacity computing. While the previous paper addressed capability based computing, the paper "Dynamic task scheduling for irregular network topologies", by M-Tahar Kechadi and Ilias K. Savvas, considers capacity based computing on a heterogeneous network and discusses a dynamic load balancing technique based on the "divide and conquer" paradigm. The goal of the technique is to minimize the response time of different active tasks in the heterogeneous computing system. It consists of two steps. At the first step, the heterogeneous computing system with irregular topology is considered as a multidimensional hyper-grid. At the second step, the tasks are distributed between nodes of the hyper-grid, dealing with one dimension at a time, and each hyper-grid balances load between its hyper-nodes.

The paper "Latency tolerance through parallelization of time in scientific applications", by Ashok Srinivasan and Namas Chandra, investigates the simulation of the molecular dynamics of nano-materials. The authors suggest an original guided simulation technique, while amounts to re-injecting the outcome of old simulations to help predict the current results. Although the simulations reported in the paper do not involve different-speed processors, the algorithm is inherently fault-tolerant, thereby opening a very interesting research perspective for heterogeneous computing platforms.

Finally, the last paper, "Workflow management and resource discovery for an intelligent grid" by Han Yu, Xin Bai and Dan C. Marinescu, provides a longer-term perspective on computational grids, which will (eventually) allow for a transparent access to heterogeneous computing resources distributed over a global network. Different grid architectures are proposed and investigated in search for the one that would be serving the best to the needs of end users. The authors mainly focus on two issues: (i) coordinated execution of computational tasks of the grid application, and (ii) locating computing resources needed for the execution.

We hope that these papers will provide an useful snapshot of current research trends in the field of heterogeneous clusters and grids. We would like to thank all the authors for their timely contributions, and all the reviewers for their time and efforts.

Alexey Kalinov
Institue for System Programming
Russain Academy of Sciences
B. Kommunisticheskaya, 25, 109004 Moscow, Russia

Alexey Lastovetsky
Department of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland

Yves Robert
LIP, Ecole Normale Supérieure de Lyon
UMR CNRS-ENS Lyon -INRIA 5668
Laboratoire LIP, F-69364 Lyon Cedex 07, France

Available online 14 June 2005