Introduction

Laurence T. Yang
Department of Computer Science,
St Francis Xavier University, Canada
E-mail: lyang@stfx.ca

Yi Pan
Department of Computer Science,
Georgia State University,
34 Peachtree Street, Suite 1450,
Atlanta, GA 30302-4110, USA
E-mail: pan@cs.gsu.edu

Biographical notes: Laurence Tianruo Yang is a Professor in Computer Science at St. Francis Xavier University, Canada. His research is mainly on high performance scientific and engineering computations with applications, design and test of embedded systems, wireless and mobile computing, pervasive computing and communications.

Yi Pan is the Chair and a Full Professor in the Department of Computer Science at Georgia State University. Dr. Pan received his BEng and MEng degrees in Computer Engineering from Tsinghua University, China, in 1982 and 1984, respectively, and his PhD degree in Computer Science from the University of Pittsburgh, USA, in 1991. Dr. Pan's research interests include parallel and distributed computing, optical networks, wireless networks, and bioinformatics. Dr. Pan has published more than 80 journal papers with 29 papers published in various IEEE journals.

With the rapid growth in computing and communication technology, the past decade has witnessed a proliferation of powerful parallel, distributed systems and networks, and an ever-increasing demand for practice of high performance computing and networking (HPCN). It has moved into the mainstream of computing, and become a key technology, which will play an important part in determining, or at least shaping, future research and development activities in many academic and industrial branches, especially when the solution of large and complex problems must cope with tight timing schedules.

Thus, high performance computing and networking deserves a separate and dedicated journal that addresses the most innovative developments in the area such as information and system architectures, grid- and web-based information management and infrastructures, data storage, management, analysis and visualisation, advanced networking with applications, scalable parallel computing, cluster and grid computing, distributed systems, and high performance scientific and engineering computing with applications.

Recognising the need for and benefit of such a publication, and with the enthusiastic support of the high performance computing and networking community, Inderscience has decided to launch a new journal. We proudly announce the inaugural issue of the *International Journal of High Performance Computing and Networking* (IJHPCN). The journal is intended to provide an outstanding channel for academics, professionals, educators, and policy makers working in the field to contribute and to disseminate innovative and important new work in high performance computing and advanced networking. This inaugural issue of the *International Journal of High Performance Computing and Networking* (IJHPCN) was prompted by the importance of high performance computing in scientific and engineering applications. One regular paper plus 12 outstanding special issue papers are included in this special issue. The special issue papers are selected from the *2nd International Workshop on Hardware/Software Support for High Performance Scientific and Engineering Computing* (SHPSEC-03) held in New Orleans, Louisiana, September 2003. The papers in this inaugural issue cover a variety of subjects addressing different aspects of high performance computing and networking.

The SHPSEC-03 special issue is mainly divided into five sections: namely system architecture, partitioning and load balancing, numerical computation, programming and environment, and computation model, respectively.

1 SYSTEM ARCHITECTURE

Switching activity and schedule length are the two most important factors that influence the energy consumption of an application executed on a VLIW (Very Long Instruction Word) processor. Considering these two factors together, Shao et al. in the first paper propose an instruction-level energy-minimisation scheduling technique to reduce the energy consumption of applications on VLIW processors. They first formally prove that this problem is NP complete. Then, three heuristic algorithms, MSAS, MLMSA, and EMSA, are proposed. The experimental results show that their heuristic algorithms give a reduction in energy compared with the traditional list scheduling approach on average.

In the second paper, Foglia et al. show how a DSS (decision support system) workload can be accelerated in the case of a shared-bus shared-memory multiprocessor, by adding simple support to the classical MESI solution for the coherence protocol. Analysis has been performed via trace driven simulation, and the operating system effects are also considered in their evaluation.

Barrier synchronisation is very important in scalable multiprocessors. As network latency rapidly approaches thousands of processor cycles and multiprocessors systems become larger and larger, conventional barrier techniques are failing to keep up with the increasing demand for efficient synchronisation. In the third paper, Fang et al. present a memory controller-based operation that optimises the barrier function of an OpenMP library. The proposed barrier achieves better performance than all other existing non-hardwired implementations, and with an improved programming interface.

In the last paper of the section, Weng et al. present the compiler transformation of OpenMP code to an ordered collection of tasks, and the compile-time as well as runtime mapping of the resulting task graph to threads for data reuse. The ordering of tasks is relaxed wherever possible so that the code may be executed in a more loosely synchronous fashion. Their current implementation uses a runtime system that permits tasks to begin execution as soon as their predecessors have completed. A comparison of the performance of two example programs in the original OpenMP form and in the code form resulting from their translation is encouraging.

2 PARTITIONING AND LOAD BALANCING

Global illumination simulation is highly critical for realistic image synthesis; without it, the rendered images look flat and synthetic. The main problem is that the simulation for large scenes is a highly time-consuming process. In the first paper of the section, Amor et al. present a uniform partitioning method in order to split the scene domain into a set of disjoint subspaces, which are distributed among the processors of a distributed memory system. Memory and communication requirements have been minimised in the parallel implementation. Good results in terms of speedup have been obtained.

The availability of low cost hardware has increased the development of distributed systems. The allocation of resources in these systems may be optimised through the use of a load balancing algorithm. The second paper by de Mello et al. presents and analyses a new load balancing algorithm that is based on a logical computer tree hierarchy. The analysis is made using results obtained by a simulator and a prototype of this algorithm. The corresponding simulations show a significant performance gain, by lowering the response times and the number of messages that pass through the communication system to perform the load balancing operations.

3 NUMERICAL COMPUTATION

In the first paper, Alberti et al. analyse the design of polylibraries, where the programs call routines from different libraries according to the characteristics of the problem and of the system used to solve it. An architecture for this type of libraries is proposed. Their aim is to develop a methodology, which can be used in the design of parallel libraries. To evaluate the viability of the proposed method, the typical linear algebra libraries' hierarchy has been considered. Experiments have been performed in different systems and with linear algebra routines from different levels of the hierarchy. The results confirm the design of polylibraries as a good technique for speeding up computations.

An object-oriented parallel toolkit has been developed by Ouarraui et al. in the second paper of the section. They are developing a parallelised implementation of the Multiview Tomography Toolbox, an iterative solver for a range of tomography problems. The authors describe their experience in parallelising a sparse matrix algorithm provided in the IML++ object-oriented numerical library. In the work, they present a parallel version of BiCGSTAB algorithm and the block-ILU preconditioner. These two algorithms are implemented in C++ using OOMPI (Object-Oriented MPI), and run on a 32-node Beowulf cluster. They also demonstrate the importance of using threads to overlap communication and computation, as an effective path to obtain improved speedup.

4 PROGRAMMING LANGUAGE

MPI support is nearly ubiquitous on high performance systems today and is generally highly tuned for performance. It would thus seem to offer a convenient 'portable network assembly language' to developers of parallel programming languages who wish to target different network architectures. Unfortunately, Bonachea et al. show that neither the traditional MPI 1.1 API nor the newer MPI 2.0 extensions for one-sided communication provide an adequate compilation target for global address space languages, and this is likely to be the case for many other parallel languages as well. Simulating one-sided communication under the MPI 1.1 API is too expensive, while the MPI 2.0 one-sided API imposes a number of significant restrictions on memory access patterns that that would need to be incorporated at the language level, as a compiler cannot effectively hide them given current conflict and alias detection algorithms.

The third paper by Roxas et al. presents an approach to specifying the different types of global reduction operations without laboriously coding the source code using the traditional textual approach. They use the visual environment provided by the system called 'Active Knowledge Studio' (AKS) to specify, modify, view, or run the specification. In general, this paper discusses how to specify global reduction operations using a language of micro-icons and shows how the visual programming environment supports the manipulation with these micro-icons.

5 COMPUTATION MODEL

A hierarchical model for parallel computations is introduced and evaluated by Qiao et al. in the first paper. This model describes the general homogeneous parallel computer systems with hierarchical parallelism and hierarchical memories (named as HPM). The HPM model consists of a hierarchy of ERAMs that cooperate with each other. The performance of HPM algorithms is discussed. The 'generalised locality' and 'memory consistency' are also proposed to analyse the algorithm performance. Their usage and examples are also given.

Previously, Adleman demonstrated that DNA (deoxyribonucleic acid) strands could be applied for dealing with solutions to an instance of the NP-complete Hamiltonian path problem (HPP). The Adleman techniques could also be used for solving the NP-complete satisfiability (SAT) problem (the first NP-complete problem). Furthermore, sticker is used for enhancing the Adleman-Lipton model. In the last paper, Guo et al. first use sticker for constructing solution space of DNA library sequences for the 3-dimensional matching problem. Then, in the Adleman-Lipton model, they propose an algorithm to remove illegal solution and find legal solution for the 3-dimensional matching problem from solution space of sticker. Finally, a simulation result for their algorithm is generated.

6 REGULAR PAPER

In a multi-computer system, a collection of processors (or nodes) work together to solve large application problems. These nodes communicate data and coordinate their efforts by sending and receiving packets through the underlying communication network. Thus, the performance of such a multi-computer system depends on the end-to-end cost of communication mechanisms. Routing time of packets is one of the key factors that are critical to the performance of multicomputers. In the only paper of the regular paper section authored by Wu et al., several enhanced sufficient conditions are given for minimal routing in 2-dimensional (2-D) meshes with faulty nodes contained in a set of disjoint faulty blocks.

ACKNOWLEDGEMENTS

Of course, the represented areas in this issue are not an exhaustive representation of the world of current high performance computing and networking. Nonetheless, they represent the rich and many-faceted knowledge that we have the pleasure of sharing with the readers. We would like to thank the authors for their excellent contributions and patience in assisting us. Finally, the dedicated work of all reviewers for these papers is also very warmly acknowledged.