Teaching Parallel Programming on Clusters

Parallel Programming—Techniques and Applications Using Networked Workstations and Parallel Computers

Barry Wilkinson and Michael Allen

431pp. $55.00

Prentice Hall, New Jersey, 1999

ISBN 0136717101

Reviewed by Rajkumar Buyya and Hai Jin

Clusters of computers have become an appealing platform for cost-effective parallel computing and especially for teaching parallel processing. An honors course in "Parallel Systems" that Rajkumar Buyya taught at Monash University focused on cluster computing. He used two books—both published by Prentice Hall in 1999—for the course. The first is the two-volume *High Performance Cluster Computing*, which Buyya edited. He used this text, along with online resources, to teach cluster computer architecture and systems issues. The second book is *Parallel Programming—Techniques and Applications Using Networked Workstations and Parallel Computers*, by Barry Wilkinson and Michael Allen, which he used to teach programming clusters using message-passing concepts.

Wilkinson and Allen's book provides an excellent discussion of various types of techniques and applications for parallel programming in cluster environments—a topic that few books successfully cover. This is a good guide for both beginners and professionals wanting to learn parallel programming on clusters. It presents the key aspects of parallel-programming concepts and generic constructs, accompanied by practical examples. Figures and flow diagrams explain each concept. The text illustrates the programs mostly in C, using generic parallel-programming constructs and popular parallel-programming interfaces such as Pthreads, Parallel Virtual Machine, and the Message Passing Interface.

The book has two parts. Part I discusses basic parallel-programming techniques as a natural extension to sequential programming. The programming techniques have been illustrated using simple programs. The techniques focus on parallel programming using message passing as well as shared memory. This part introduces the basic concept of parallel programming and the types of computer systems. The authors highlight two basic parallel programming paradigms: *message passing computing*, using PVM and MPI, and *shared memory programming*, with standard UNIX processes and Pthread. The authors also discuss three different parallel computation models, with detailed examples: embarrassingly parallel computation, pipelined computation, and synchronous computation. They explore fundamental techniques for parallel programming, such as partitioning, divide and conquer**,** load balancing, and termination detection.

Part II addresses problem-specific algorithms in both nonnumeric and numeric domains. The book covers numeric domain topics such as sorting, matrix multiplications, linear equations, partial differential equations, and Jacobi iterations and their parallel implementations using message-passing paradigms. The nonnumeric problems covered in this part include the imaging-processing searching and optimization applications. The imaging-processing application is particularly suitable for parallelization and has significant potential for parallel programming projects. The section covers the FFT (Fast Fourier Transformation) application in the context of image processing. The FFT techniques are useful for many application areas such as including signal processing and voice recognition. Searching and optimization are two very important issues with various applications including commerce, banking, and industry. The text presents branch-and-bound methods as the fundamental searching and optimization technique, and discusses genetic algorithm with underlying techniques and various parallelization approaches in the context of searching and optimization. Other techniques, such as the hill-climbing technique, are applied to a banking optimization problem.

The book has extensive appendices providing a quick index to basic PVM, MPI and Pthread routines. The appendix briefly outlines several theoretical (algorithm design for abstract machine) models for parallel computations, including Parallel Random-Access Machine, Bulk-Synchronous Parallel, and LogP (Latency, overhead, gap, and Processor) models. Each chapter ends with a clear summary of the concepts and terms used, an extensive guide to further reading materials, a state-of-the-art bibliography for each topic, and well-designed problems for use by instructors. The authors have also created an excellent Web resource that offers presentation slides, program source codes, and instructional materials for using PVM, MPI, and Pthread software. All these tools enhance the experience of teaching parallel programming. (See the "Cluster Computing Resources" sidebar.)

This book would have been more valuable and helpful had it come with a CD-ROM containing these parallel programming tools and illustrative demo applications. All in all, however, we found Wilkinson and Allen's book quite useful in our teaching. We do not hesitate to recommend it to anyone who is serious about teaching or learning parallel programming on clusters.

Rajkumar Buyya: School of Computer Science and Software Eng' Monash University, Melbourne, Australia
rajkumar@csse.monash.edu.au

Hai Jin: Department of Electrical Engineering' University of Southern California, Los Angeles, California, USA,
hjin@ceng.usc.edu

Cluster Computing Resources on the Internet:

Cluster Computing Courseware @ Monash: http://www.csse.monash.edu.au/~rajkumar/csc433/
Barry and Allen Book Resource Centre: http://www.coe.uncc.edu/~abw/parallel/par_prog
Cluster Computing Info. Centre: http://www.buyya.com/cluster
IEEE Task Force on Cluster Computing (TFCC) – http://www.ieeetfcc.org
IEEE Intl. Conference on Cluster Computing: http://www.clustercomp.org
IEEE Intl. Symposium on Cluster Computing and the Grid: http://www.ccgrid.org
TFCC and TOP500 Initiative on TopClusters List – http://www.TopClusters.org