Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

More information about this series at http://www.springer.com/series/7592

Roman Trobec • Boštjan Slivnik
Patricio Bulić • Borut Robič

Introduction to Parallel Computing

From Algorithms to Programming on State-of-the-Art Platforms

Springer

Roman Trobec
Jožef Stefan Institute
Ljubljana, Slovenia

Patricio Bulić
University of Ljubljana
Ljubljana, Slovenia

Boštjan Slivnik
University of Ljubljana
Ljubljana, Slovenia

Borut Robič
University of Ljubljana
Ljubljana, Slovenia

ISSN 1863-7310 ISSN 2197-1781 (electronic)
Undergraduate Topics in Computer Science
ISBN 978-3-319-98832-0 ISBN 978-3-319-98833-7 (eBook)
https://doi.org/10.1007/978-3-319-98833-7

Library of Congress Control Number: 2018951197

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To all who make our lives worthwhile.

Preface

This monograph is an overview of practical parallel computing and starts with the basic principles and rules which will enable the reader to design efficient parallel programs for solving various computational problems on the state-of-the-art computing platforms.

The book too was written in parallel. The opening Chap. 1: "Why do we need Parallel Programming" has been shaped by all of us during instant communication immediately after the idea of writing such a book had cropped up. In fact, the first chapter was an important motivation for our joint work. We spared no effort in incorporating of our teaching experience into this book.

The book consists of three parts: Foundations, Programming, and Engineering, each with a specific focus:

- Part I, **Foundations**, provides the *motivation* for embarking on a study of parallel computation (Chap. 1) and an *introduction* to parallel computing (Chap. 2) that covers parallel computer systems, the role of communication, complexity of parallel problem-solving, and the associated principles and laws.
- Part II, **Programming**, first discusses *shared memory* platforms and OpenMP (Chap. 3), then proceeds to *message passing* library (Chap. 4), and finally to *massively parallel* processors (Chap. 5). Each chapter describes the methodology and practical examples for immediate work on a personal computer.
- Part III, **Engineering**, *illustrates* parallel solving of computational problems on three selected problems from three fields: *Computing the number* π (Chap. 6) from mathematics, *Solving the heat equation* (Chap. 7) from physics, and *Seam carving* (Chap. 8) from computer science. The book concludes with some final remarks and perspectives (Chap. 9).

To *enable readers to immediately start gaining practice* in parallel computing, **Appendix A** provides hints for making a personal computer ready to execute parallel programs under Linux, macOS, and MS Windows.

Specific contributions of the authors are as follows:

- *Roman Trobec* started the idea of writing a practical textbook, useful for students and programmers on a basic and advanced levels. He has contributed Chap. 4: "MPI Processes and Messaging", Chap. 9: "Final Remarks and Perspectives", and to chapters of Part III.

- *Boštjan Slivnik* has contributed Chap. 3: "Programming Multi-core and Shared Memory Multiprocessors Using OpenMP". He has also contributed to Chap. 1 and chapters of Part III.
- *Patricio Bulić* has contributed Chap. 5: "OpenCL for Massively Parallel Graphic Processors" and Chap. 8: "Engineering: Parallel Implementation of Seam Carving". His contribution is also in Chap. 1 and chapters of Part III.
- *Borut Robič* has coordinated our work and cared about the consistency of the book text. He has contributed Chap. 2: "Overview of Parallel Systems" and to Chap. 1: "Why do we need Parallel Programming".

The spectrum of book topics ranges from implementations to efficient applications of parallel processors on different platforms. The book covers shared memory many-core processors, shared memory multi-core processors, and interconnected distributed computers. Chapters of Parts I and II are quite independent and can be read in any order, while chapters of Part III are related to previous chapters and are intended to be a final reading.

The target audience comprises undergraduate and graduate students; engineers, programmers, and industrial experts acting in companies that develop software with an intention to add parallel capabilities for increased performance; research institutions that develop and test computationally intensive software with parallel software codes; and universities and educational institutions that teach courses on parallel computing. The book may also be interesting and useful for the wider public in the part where basic principles and benefits of parallel approaches are presented.

For the readers who wish to be promptly updated with current achievements in the field of parallel computing, we will maintain this information on the book web page. There, also a pool of questions and homeworks will be available and maintained according to experiences and feedbacks from readers.

We are grateful to all our colleagues who have contributed to this book through discussions or by reading and commenting parts of the text, in particular to Matjaž Depolli for his assistance in testing the exemplar programs, and to Andrej Brodnik for his fruitful suggestions and comments.

For their support of our work, we are indebted to the Jožef Stefan Institute, Faculty of Computer and Information Science of the University of Ljubljana, and the Slovenian Research Agency.

Ljubljana, Slovenia
June 2018

Roman Trobec
Boštjan Slivnik
Patricio Bulić
Borut Robič

Contents

Part I Foundations

1 Why Do We Need Parallel Programming 3
1.1 Why—Every Computer Is a Parallel Computer 3
1.2 How—There Are Three Prevailing Types of Parallelism 4
1.3 What—Time-Consuming Computations Can Be Sped up 5
1.4 And This Book—Why Would You Read It? 7

2 Overview of Parallel Systems 9
2.1 History of Parallel Computing, Systems and Programming 9
2.2 Modeling Parallel Computation 11
2.3 Multiprocessor Models 13
2.3.1 The Parallel Random Access Machine 13
2.3.2 The Local-Memory Machine 17
2.3.3 The Memory-Module Machine 18
2.4 The Impact of Communication 18
2.4.1 Interconnection Networks 19
2.4.2 Basic Properties of Interconnection Networks 19
2.4.3 Classification of Interconnection Networks 22
2.4.4 Topologies of Interconnection Networks 25
2.5 Parallel Computational Complexity 31
2.5.1 Problem Instances and Their Sizes 31
2.5.2 Number of Processing Units Versus Size of Problem Instances 32
2.5.3 The Class NC of Efficiently Parallelizable Problems 33
2.6 Laws and Theorems of Parallel Computation 36
2.6.1 Brent's Theorem 36
2.6.2 Amdahl's Law 37
2.7 Exercises 42
2.8 Bibliographical Notes 44

Part II Programming

3 Programming Multi-core and Shared Memory Multiprocessors Using OpenMP 47
3.1 Shared Memory Programming Model 47
3.2 Using OpenMP to Write Multithreaded Programs 49
3.2.1 Compiling and Running an OpenMP Program 50
3.2.2 Monitoring an OpenMP Program 52
3.3 Parallelization of Loops 53
3.3.1 Parallelizing Loops with Independent Iterations 54
3.3.2 Combining the Results of Parallel Iterations 62
3.3.3 Distributing Iterations Among Threads 72
3.3.4 The Details of Parallel Loops and Reductions 76
3.4 Parallel Tasks 78
3.4.1 Running Independent Tasks in Parallel 78
3.4.2 Combining the Results of Parallel Tasks 82
3.5 Exercises and Mini Projects 84
3.6 Bibliographic Notes 86

4 MPI Processes and Messaging 87
4.1 Distributed Memory Computers Can Execute in Parallel 87
4.2 Programmer's View 88
4.3 Message Passing Interface 89
4.3.1 MPI Operation Syntax 92
4.3.2 MPI Data Types 93
4.3.3 MPI Error Handling 95
4.3.4 Make Your Computer Ready for Using MPI 95
4.3.5 Running and Configuring MPI Processes 95
4.4 Basic MPI Operations 98
4.4.1 `MPI_INIT (int *argc, char ***argv)` 98
4.4.2 `MPI_FINALIZE ()` 98
4.4.3 `MPI_COMM_SIZE (comm, size)` 98
4.4.4 `MPI_COMM_RANK (comm, rank)` 98
4.5 Process-to-Process Communication 99
4.5.1 `MPI_SEND (buf, count, datatype, dest, tag, comm)` 100
4.5.2 `MPI_RECV (buf, count, datatype, source, tag, comm, status )` 101
4.5.3 `MPI_SENDRECV (sendbuf, sendcount, sendtype, dest, sendtag, recvbuf, recvcount, recvtype, source, recvtag, comm, status)` 103
4.5.4 Measuring Performances 104

4.6 Collective MPI Communication . . . 107
4.6.1 MPI_BARRIER (comm) . . . 107
4.6.2 MPI_BCAST (*inbuf*, incnt, intype, root, comm) . . . 108
4.6.3 MPI_GATHER (inbuf, incnt, intype, outbuf, outcnt, outtype, root, comm) . . . 108
4.6.4 MPI_SCATTER (inbuf, incnt, intype, outbuf, outcnt, outtype, root, comm) . . . 109
4.6.5 Collective MPI Data Manipulations . . . 110
4.7 Communication and Computation Overlap . . . 114
4.7.1 Communication Modes . . . 115
4.7.2 Sources of Deadlocks . . . 117
4.7.3 Some Subsidiary Features of Message Passing . . . 122
4.7.4 MPI Communicators . . . 123
4.8 How Effective Are Your MPI Programs? . . . 128
4.9 Exercises and Mini Projects . . . 129
4.10 Bibliographical Notes . . . 131

5 OpenCL for Massively Parallel Graphic Processors . . . 133
5.1 Anatomy of a GPU . . . 133
5.1.1 Introduction to GPU Evolution . . . 134
5.1.2 A Modern GPU . . . 138
5.1.3 Scheduling Threads on Compute Units . . . 139
5.1.4 Memory Hierarchy on GPU . . . 142
5.2 Programmer's View . . . 145
5.2.1 OpenCL . . . 145
5.2.2 Heterogeneous System . . . 146
5.2.3 Execution Model . . . 146
5.2.4 Memory Model . . . 148
5.3 Programming in OpenCL . . . 150
5.3.1 A Simple Example: Vector Addition . . . 150
5.3.2 Sum of Arbitrary Long Vectors . . . 173
5.3.3 Dot Product in OpenCL . . . 176
5.3.4 Dot Product in OpenCL Using Local Memory . . . 180
5.3.5 Naive Matrix Multiplication in OpenCL . . . 186
5.3.6 Tiled Matrix Multiplication in OpenCL . . . 189
5.4 Exercises . . . 195
5.5 Bibliographical Notes . . . 195

Part III Engineering

6 Engineering: Parallel Computation of the Number π 199
6.1 OpenMP 202
6.2 MPI 204
6.3 OpenCL 208

7 Engineering: Parallel Solution of 1-D Heat Equation 211
7.1 OpenMP 215
7.2 MPI 216

8 Engineering: Parallel Implementation of Seam Carving 223
8.1 Energy Calculation 225
8.2 Seam Identification 226
8.3 Seam Labeling and Removal 229
8.4 Seam Carving on GPU 232
8.4.1 Seam Carving on CPU 233
8.4.2 Seam Carving in OpenCL 235

9 Final Remarks and Perspectives 241

Appendix: Hints for Making Your Computer a Parallel Machine 243

References 251

Index 253