Introduction to Parallel Computing

The constantly increasing demand for more computing power can seem impossible to keep up with. However, multicore processors capable of performing computations in parallel allow computers to tackle ever larger problems in a wide variety of applications. This book provides a comprehensive introduction to parallel computing, discussing both theoretical issues such as the fundamentals of concurrent processes, models of parallel and distributed computing, and metrics for evaluating and comparing parallel algorithms, as well as practical issues, such as methods of designing and implementing shared- and distributed-memory programs, and standards for parallel program implementation, in particular MPI and OpenMP interfaces.

Each chapter presents the basics in one place, followed by advanced topics, allowing both novices and experienced practitioners to quickly find what they need. A glossary and more than 80 exercises with selected solutions aid comprehension. The book is recommended as a text for advanced undergraduate or graduate students and as a reference for practitioners.

Zbigniew J. Czech is Professor of Computer Science at Silesian University of Technology, Gliwice, Poland. His research interests include computer programming, design and analysis of algorithms, and parallel computing, on which he has more than 45 years of experience lecturing and conducting research. He has served as a research fellow at the University of York and the University of Canterbury in the United Kingdom, and has lectured at numerous universities in Poland and elsewhere, including the University of California–Santa Barbara, Indiana University-Purdue University, and the University of Queensland.

INTRODUCTION TO PARALLEL COMPUTING

Zbigniew J. Czech
Silesian University of Technology

CAMBRIDGE UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi - 110002, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107174399

First published 2016

Printed in the United States of America by Sheridan Books, Inc.

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Czech, Zbigniew J.
Title: Introduction to parallel computing / Zbigniew J. Czech, Silesia University of Technology.
Description: Cambridge, United Kingdom ; New York, NY, USA : Cambridge University Press, 2016. | Includes bibliographical references and index.
Identifiers: LCCN 2016051952 | ISBN 9781107174399 (hardback : alk. paper)
Subjects: LCSH: Parallel processing (Electronic computers)
Classification: LCC QA76.58.C975 2016 | DDC 004/.35 – dc23
LC record available at https://lccn.loc.gov/2016051952

ISBN 978-1-107-17439-9 Hardback

Contents

List of Figures

List of Tables

Preface

Solving contemporary scientific and technology problems requires the use of computers with a high speed of computation. Over the last 60 years the rate of this speed has increased 16 trillion (10^{12}) times. In the 1950s the speed of computation of a Univac 1 computer was about 1 kflop/s (flop denotes a floating-point operation), and in 2015 China's supercomputer Tianhe-2 (Milky Way-2), which contained 3 120 000 cores working in parallel, achieved the computation speed of more than 33 Pflop/s (petaflop stands for one quadrillion, 10^{15}, floating-point operations). Despite a significant increase in computational capabilities, researchers simplify models of considered problems because their numerical simulation takes too long. The demand for more and more computing power has increased and it is believed that this trend will continue in the future. There are several reasons behind this trend. Models of investigated phenomena and processes have become more complex and larger amounts of data are being processed. The requirements regarding accuracy of results also grow, which entails a higher resolution of models being developed. The fields in which, through large computing power, significant results have been achieved include: aeronautics, astrophysics, bioinformatics, chemistry, economics and trade, energy, geology and geophysics, materials science, climatology, cosmology, medicine, meteorology, nanotechnology, defense, and advanced engineering.

For example, in the U.S. National Aeronautics and Space Agency (NASA), simulation problems related to research missions conducted by space shuttles have been investigated [50]. A parallel computer SGI Altix with 10 240 processors, consisting of 20 nodes each holding 512 processors, installed in the J. Ames Center allowed for simulation of a pressure distribution around a space shuttle during its flight. A package of computational fluid dynamics used for this goal was a tool for designing geometry of the parts of a space shuttle, that is, launchers and orbital units. Another group of issues resolved in the NASA research centers concerned jet drive units. One of the tasks was to simulate a flow of liquid fuel supplied to a space shuttle main engine by a turbine pump ([31], sect. 2.4).

In order to improve aircraft performance and safety, NASA conducts research in new aircraft technologies. One of the objectives is the accurate prediction of aerodynamic and structural performance for rotorcraft designed for civil and military applications. New physics-based computational tools to predict rotorcraft flowfields by

using the non-linear, three-dimensional, Navier-Stokes equations have been developed. The tools have enabled high-fidelity simulations of a UH-60 Blackhawk helicopter rotor in high-speed forward flight [68, 69]. All simulations were run on the Pleiades petascale supercomputer [51] installed in the NASA Advanced Supercomputing Division at Ames Research Center. Each run of simulation used 1536–4608 cores included in Pleiades Westmere nodes.

After the final shuttle mission in 2011, NASA began to concentrate its efforts on human space exploration in and beyond low-Earth orbit. The efforts include design of the Space Launch System (SLS), the next generation heavy lift launch vehicle with its first developmental flight planned in late 2017. Pleiades plays an important role in producing comprehensive computational fluid dynamics (CFD) simulations for design analyses of SLS vehicle. The analyses are used to predict the aerodynamic performance, load, and pressure signatures for design variations of both crew and cargo vehicles. Up to 2012, more than 3300 cases for seven different SLS designs have been simulated on Pleiades using three independent CFD flow solvers: OVERFLOW, USM3D, and NASA's Cart3D. Best practices for simulating launch vehicle ascent using those solvers were established during the Constellation Program. Simulations were run on either the Columbia or Pleiades supercomputer using 64 to 96 processors [218].

A significant impact on human development has been research in climatology. This will help to answer such fundamental questions as: Is the observed recently average Earth temperature rise a sign of ongoing global warming and a looming climate catastrophe? and also: Is the temperature growth due to natural reasons or is it the result of the greenhouse effect[1] caused by increased emissions of carbon dioxide (CO_2) and particulate matter into the atmosphere, resulting from human activities including burning coal and other fossil fuels? Answers to these questions are sought by numerical simulation of climate. For this study a program based on the Community Climate System Model (CCSM3) [80], maintained by the National Center for Atmospheric Research (NCAR), was used. It contained four components that described the state of the atmosphere, land, oceans, and ice caps. The components of a program were executed in parallel on disjoint sets of various number of processors, which exchanged data describing, among other parameters, flows of masses and energy. The CCSM3 version of a program allowed for simulation of the globe climate with 75 km resolution (distance of points in a numeric grid[2]) over several hundred years [395]. Simulations were capable of analyzing "what if ?" cases; for example, one can predict to what extent sea levels will rise as a result of melting ice, if the amount of CO_2 emitted into the atmosphere doubles.

The results presented in [112] attained on the Cray X1E and XT4 computers and IBM p575 and p690 Cluster indicated that the Community Atmosphere Model component could simulate a state of the atmosphere over a span up to several tens of years, in a single day of computation by making use of several hundred processors. Scalability of the component to simulate a state of oceans (Parallel Ocean Program) was better. By making use of 500–1000 processors, results of

[1] This issue of how hot the world will be due to the greenhouse effect was found in *Science* journal as one of the 25 great puzzles of science in 2005 [117] (see also [340]).

[2] Simulation results for shorter periods of time with 10–20 km resolution were obtained using the Japanese Earth Simulator parallel computer, see for example [271].

simulation for a period of several hundred years could be obtained in a single day of calculation.

Since CCSM3's release in 2007, the work to improve and adapt its core to be implemented on petaflop computers with a number of processors from 100 to 200 thousands has been continued [112]. In 2010 NCAR released the fourth version of CCSM (CCSM4) [150], www.cesm.ucar.edu/model/cssm4.0/, and in the same year the successor to CCSM, called the Community Earth System Model version 1 (CESM1), www.cesm.ucar.edu/models/cesm1.0, [254], was published as a unified code release that included CCSM4 code as a subset.

Based on the CCSM4_alpha code version, high-resolution, century-scale simulations of the Earth's climate were run on the teraflop Cray XT4 and XT5 supercomputers [98]. The resolutions adopted for experiments were 50 km for the atmosphere and land surface, and 20 km for ocean/sea-ice. The tests revealed that the development version of CCSM4 was capable to achieve 2.3 simulated years per day on the Cray XT5 utilizing 5844 computing cores. One of the enhancements in the CESM1 model relative to the CCSM4 was the inclusion of an atmospheric component that extended in altitude to the lower thermosphere. This atmospheric model, known as the Whole Atmosphere Community Climate Model, allowed simulation of the climate change from 1850 to 2005 [261]. The computations were performed on the IBM Bluefire supercomputer. Using 192 POWER6 processors, the model was capable of generating approximately 4.5 simulated years per day.

Over the last decade, several other models connected with the Earth's climate have emerged. One of them is the Non-hydrostatic Icosahedral Atmospheric Model (NICAM) developed mainly at the Japan Agency for Marine-Earth Science and Technology, the University of Tokyo, and the RIKEN research institute. Recently, NICAM has produced the global atmosphere simulation with an unprecedented horizontal resolution of 870 m. The simulation has been executed with a performance of 230 Tflop/s for 68 billion grid cells while using 20 480 nodes (163 840 cores) of the K computer [334, 223, 405].

Powerful parallel computers have played a significant role in human genome reading. In 1990, the U.S. and UK initiated the Human Genome Project (HGP), which aimed to decipher the information contained in the human DNA (Deoxyribonucleic Acid). The project was joined by China, France, Japan, and Germany. Parallel work on the description of the human genome has been undertaken by American Craig Venter and his employees in his biotechnology enterprise The Institute for Genetic Research, transformed later into Celera Genomics. The research teams initially competed against each other, but over time the race transformed into cooperation, so that studies were not duplicated, which made it possible to control faithfulness of genome reading. In May 2001, the teams published results of their research independently [203, 390] and in April 2003 they announced completion of the projects. Studies have shown that the human genome containing more than 3 billion base pairs has 30 000 genes coding proteins and RNA molecules. They represent approximately 1.5% of the total DNA. The rest is known as noncoding DNA ("junk DNA") accumulated during man's evolution [400].

To decipher mysteries of DNA Celera Genomics used a parallel computer of cluster architecture consisting of ten 4-processor SMP (Symmetric Multiprocessor) nodes, each with 4 GB of memory (Compaq ES40), and a 16-processor NUMA (Nonuniform Memory Access) computer equipped with memory of capacity of

64 GB (Compaq GS160). Execution of a DNA sequencing program required approximately 20 000 hours [390]. The computational infrastructure of the HGP consortium comprised a computing server Compaq ES40 consisting of 27 nodes, each with 108 processors, and a file server with external memory of capacity of 1 TB [203].

Description of human genome greatly accelerated development of bioinformatics, a separate discipline of research and applications. Contemporary bioinformatics represents a convergence of various areas, including modeling of biological phenomena, genomics, biotechnology. In the last few decades an enormous amount of biological data has been collected, which generated demand for novel algorithms and tools to analyze and decipher the complexity of such large data. Attaining these objectives requires high-performance computing and advanced storage capabilities. A review of applications of supercomputers in sequence analysis and genome annotation, two of the emerging and most important branches of bioinformatics, is given in [115].

Despite gathering a vast amount of knowledge about the human genome, a number of questions have not been answered yet. The role of pseudo-genes that have a form of genes but do not encode proteins is not known. What is the meaning of "discontinuity" of genes associated with presence of the so-called introns? And does, however, the noncoding DNA contain useful information? The further studies of the genome will be continued, because its complete understanding is important for many fields. For example, in medicine it will give us an opportunity to diagnose gene-based hereditary diseases and conduct gene therapy. Genetic tests already allow for determining paternity and are used in criminology. In the future, genetic knowledge will facilitate individual adjustment of drugs to the needs of patients, which will extend people's life expectancy and improve their quality of life.

It is likewise important to know the genomes of plants. By their modifications one can breed new varieties of plants characterized by higher fertility and resistance to drought and pests. For example, the results of decoding a maize genome employing a Blue Gene/L computer [142, 143] with 1024 processors were reported in [211].

In recent decades an increasing role of computers in research has been observed. Some scientific discoveries, such as the aforementioned reading of the human genome, have been made due to substantial power of computation. The use of advanced methods and means of computing to solve complex problems is a domain of computational science. The third pillar of this field of science, in addition to theory and experiment, is computer modeling and simulation of large-scale phenomena. The importance of computational science in the context of competitiveness and prosperity of the society was a subject of the U.S. President's Information Technology Advisory Committee report [312] (see also [291]).

Parallel computers are expensive installations with a cost of tens of millions of dollars. Therefore, for a long time only government research institutions were equipped with them. But recently they have begun to appear in industrial sectors of high income. Oil companies benefit from high-end computers by using them to manage effectively the existing oil and gas deposits [220] (access.ncsa.uiuc.edu/Stories/oil) and to search for new ones. Car manufacturers use sophisticated software to simulate a vehicle collision with obstacles [338] and to simulate flow of air around a car body. This helps to increase safety and efficiency of the proposed designs and to reduce the time of introducing new car models into sale [275]. Pharmaceutical companies use high-speed computers for designing new drugs [28, 64, 119]. The faster a drug is discovered the sooner it can be patented and brought to the market.

In the 1970s aircraft manufacturers could simulate a pressure distribution around a single wing of an aircraft. Currently such a simulation is possible for the entire structure of an aircraft [393, 273]. As a consequence, the use of expensive wind tunnels is increasingly rare. Application of high performance computing equipment enables manufacturers not only to increase profits and reduce production costs, but also to gain an advantage over competitors.

Other applications of parallel computers include numerical weather prediction [345, 254], www.science.gov/topicpages/w/wrf+weather+research.html, and forecasting natural disasters [201], such as earthquakes [104], earthquake.usgs.gov/research/, volcanic eruptions, tsunami waves [32, 262], nctr.pmel.noaa.gov, hurricanes [73], www.nhc.noaa.gov, tornados [404], and tropical cyclones [326].

As a result of ongoing efforts to boost performance of processors, complexity of integrated circuits (IC) and their degree of integration increase. Enhancement of this performance by improving the technology used so far has recently faced insurmountable obstacles. Packing more and more transistors into smaller and smaller volumes makes the width of paths inside IC components approach the size of atoms. The rise of total intensity of currents flowing between a large number of transistors causes an increase in the amount of dissipated heat. Collection of this heat from very small volumes becomes very difficult. Boosting the speed of computation by raising the clock frequency has its limits due to delays of signals transmitted along the paths in a chip. All these obstacles have given rise to construction of multicore processors consisting of a number of cores contained in a single chip of a slightly larger size compared with a conventional one. Since each core executes an independent instruction stream, computations in a multicore processor are parallel in nature. Multicore processors are becoming more and more popular. They are used in general purpose and personal computers, computing servers, embedded systems, gaming consoles, etc. Along with the popularization of multicore processors, the importance of issues related to the design and implementation of parallel programs for these types of processors—which are discussed in this book—will grow.

The speed of modern supercomputers due to parallel operation of more than 3 million interconnected processors[3] is now of the order of quadrillion operations per second. As a result of technological advances this power systematically increases. Breaking the next barrier of computation speed of 1 Eflop/s (exaflop denotes one quintillion, 10^{18}, operations) will facilitate the solution of key issues in the study of health, prosperity, security, and the development of mankind. In medicine and biological sciences it will be possible to simulate molecular phenomena and to explain complex processes of protein folding [102]. Simulation of electromagnetic, thermal, and nuclear interactions between particles in a variable magnetic field assists in the development of devices in which one can conduct controlled thermonuclear fusion (www.iter.org). Production of such devices on an industrial scale would be a breakthrough in solving the energy problems of the world, as well as in spacecraft propulsion technology. Fast parallel computers with memories of large capacities give opportunity to explore huge databases in world trade and economy. This allows for better understanding of phenomena and economic trends for the benefit of people welfare. In the field of defense—following adoption by a group of states of the Comprehensive Nuclear-Test-Ban Treaty in 1996—performing time-consuming simulation has

[3] See the description of Tianhe-2 supercomputer on p. 296.

become essential to maintain readiness of strategic weapons stockpiles. Fast computations may enhance safety through the use of advanced cryptographic and cryptanalytic methods for encryption and decryption of messages in real-time employing increasingly complex codes.

This book is devoted to the issues concerning implementation of parallel computing. In particular, it discusses the stages of analysis, design, and implementation of parallel programs. The book is recommended as a text for advanced undergraduate or graduate students. It can also be helpful for practitioners who are involved in parallel computing, such as programmers, system designers, and for all those interested in the subject. The reader should be familiar with programming in at least one of the high-level languages, for example C/C++, as well as with the basics of algorithms.

The book is a result of the experience I have gathered over the past dozen years conducting research and giving lectures on parallel computing for students at the Silesian University of Technology, Gliwice, and the University of Silesia, Sosnowiec, in Poland. A large number of valuable comments and suggestions on the first edition of the book were conveyed to me by my colleagues: Agnieszka Debudaj-Grabysz, Sebastian Deorowicz, Wojciech Mikanik, Rafał Skinderowicz, Jacek Widuch, and Wojciech Wieczorek. The doctoral students: Mirosław Błocho, Sergiusz Michalski, and Jakub Nalepa gave me useful remarks to the second edition of the book. Jakub Nalepa prepared implementations of several parallel programs that are enclosed in the exercise solutions, and Jakub Rosner helped in updating the GPUs description. I would like to express my deep appreciation to all people mentioned above for their time and commitment in helping to improve the content and form of the two Polish editions, and the present English edition of the book.

I also thank the staff of the following Polish computing centers where computations of the project were carried out: Academic Computer Center, Gdańsk (TASK); Wrocław Center for Networking and Supercomputing (WCSS); Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw (ICM); Academic Computer Center CYFRONET AGH, Kraków; Poznań Supercomputing and Networking Center (PCSS).

The book consists of seven chapters. In Chapter 1 the concepts of a concurrent process and thread as units executed under supervision of the operating system are introduced. The ways processes communicate with each other, the issue of proving correctness of concurrent programs and selected problems in concurrent programming are also presented. Chapter 2 is devoted to basic models of parallel computation. The details of the PRAM (Parallel Random Access Machine) and of the network models are discussed. Chapter 3 focuses on the elementary parallel algorithms and methods of their evaluation using selected metrics, such as parallel running time, speedup, cost, and efficiency. The problem of scalability of parallel algorithms is formulated, and related to it the Amdahl's law, Gustafson–Barsis's law, and Karp–Flatt metric are described. Chapter 4 is devoted to the methods of parallel algorithms design. The basic steps of design are considered, in particular decomposition of a computational problem into tasks, analysis of computation granularity, minimizing the parallel algorithm cost, and assigning tasks to processors. Chapter 5 deals with parallel computer architectures. It provides a short description of structures of processor arrays, multiprocessors with shared and distributed memory, computing clusters, and computers with unconventional architectures. An overview of interconnection networks is also given. Chapters 6 and 7 focus on principles of parallel program

design for message passing and shared memory models. These principles are illustrated with examples of programs created by employing the MPI (Message Passing Interface) library and OpenMP (Open Multiprocessing) interface. Each chapter of the book is supplemented with exercises that permit the reader better understanding and assimilation of the content presented in a chapter. Solutions to selected exercises, and a glossary of parallel computing terms appear at the end of the book.

Gliwice, Poland, May 2016 *Zbigniew J. Czech*