A Practical Approach to High-Performance Computing

Sergei Kurgalin • Sergei Borzunov

A Practical Approach to High-Performance Computing

Springer

Sergei Kurgalin
Department of Digital Technologies
Voronezh State University
Voronezh, Russia

Sergei Borzunov
Department of Digital Technologies
Voronezh State University
Voronezh, Russia

ISBN 978-3-030-27557-0 ISBN 978-3-030-27558-7 (eBook)
https://doi.org/10.1007/978-3-030-27558-7

Preface

This book discusses the fundamentals of high-performance computing. The study and mastery of these are essential for solving current problems in fundamental and applied science, and will allow the growing demands of researchers for increased processing speed of large data volumes to be satisfied. In this book, we have attempted to combine visualization, comprehensibility, and rigor of the presentation of the material, with an orientation towards practical applications and training to solve real computing problems.

The book is based on the authors' experience of many years of teaching at Voronezh State University of courses with titles such as "Parallel Programming," "Technologies for Parallel Computing," "Parallel and Grid Technologies," and "Parallel Data-Processing Algorithms" for bachelor's and master's degree students, as well as a course entitled "Information and Communication Technologies" for upgrading the qualifications of university teachers and research fellows. Since 2002, when it launched its first high-performance parallel computer cluster, the university has been providing teaching in the field of supercomputing.

The book is intended for a wide range of levels of readers, from juniors beginning to familiarize themselves with the subject to master's degree students, postgraduates, research fellows, and specialists.

The material of the book has been thoroughly selected and consistently presented to give an idea of various aspects of high-performance computation. The book contains basic theoretical insight into the methods of parallel programming and information about the techniques for parallelizing.

The material is presented at a rather rigorous mathematical level; all statements (lemmas and theorems) are provided with proofs, or, in rare instances, with references to the specialized literature on the topic being discussed.

Each chapter begins with a theoretical part, where the relevant terminology is introduced along with the basic theoretical results and methods of parallel programming. Apart from the basic definitions, theoretical provisions, and computing models, the text contains instructions for their use when developing parallel programs. Furthermore, several standard problems are analyzed in detail. When an example is provided in the text, its end is marked with a square. Each chapter

is concluded by a list of test questions for checking and consolidation of the theoretical material; then follow problems to be solved in the lecture hall or computer classroom. These problem can also be used for individual study. The exercises differ considerably in their complexity. The most difficult ones are marked with an asterisk in front of the number.

Many problems are provided with answers, hints, and solutions, including sample program code. The texts of programs demonstrating the solutions of both standard and more complex problems are usually provided in full and unabridged form, which is especially important for first acquaintance with the book's material. The programs are written with plenty of explanatory comments.

There exist two main approaches to the programming of modern computing systems: multithreading-based parallelizing in shared-memory systems and the application of message-passing technologies in distributed systems. Both of these approaches are discussed in the book.

Most of the parallel programs used as examples are supplemented with speedup curves for the computational part of the program code as a function of the number of threads in a shared-memory system. The speedup values were calculated for the supercomputer at Voronezh State University. Visualization of the dependence of the speedup on the number of threads allows students to gain a deeper understanding of the operating principles of parallel algorithms.

The examples and problems provided in the book pay much attention to ensuring the required computational accuracy. In all cases, the computational error bounds required from the program are strictly defined.

The inclusion of reference information in the book allows the reader to consult the specialized literature less often than would otherwise be necessary. The appendices accomplish the methodological task of providing students with the skills required for working with the command shell responsible for communication between the user and the supercomputer system. The book also provides training materials covering the construction and analysis of parallel algorithms. As a result, working with the book should require very little resort to additional resources. The inclusion of extensive reference material and recommendations on the installation of parallel environments and the starting of programs on a supercomputer allows the reader to obtain skills in parallel programming even with minimal knowledge in the field of information and communication technologies. It is only necessary to know the basic constructs of the algorithmic language C, which is used to realize the algorithms and describe the problem solutions presented in the book. The C language was chosen because this language, on the one hand, is widely known and, on the other hand, is one of the main languages for supercomputer computations.

The book includes many illustrations visualizing the objects being studied and their interconnections.

The book consists of six chapters and four appendices.

Chapter 1 offers an overview of the basic methods of parallel programming and discusses the classification of computing system architectures.

Chapter 2 looks into the basic topologies of data-passing networks. It is shown that the network topology has a considerable influence on the performance of the computing system as a whole.

Chapter 3 is devoted to the basic laws of parallel computing. Within the PRAM model, the quantitative characteristics of the efficiency of parallel algorithms are discussed. Here, the notion of an "operations–operands" graph is introduced and Bernstein's conditions are formulated.

The next two chapters contain descriptions of widely used parallel programming technologies.

Chapter 4 is devoted to a description of the distributed-memory-system programming technology MPI. The essential functions of MPI, the operations of pairwise message exchange in blocking and nonblocking modes, and collective data-passing operations are discussed.

Chapter 5 considers the shared-memory computing system programming environment OpenMP, and the computing model used when working with this environment.

Chapter 6 contains implementations of basic parallel algorithms using the OpenMP technology as an example. Algorithms for array element summation, data sorting, basic matrix operations, and Monte Carlo static tests are discussed here.

Appendices A, B, C offer reference material, including methods for analysis of algorithms, a description of working with the Linux shell, and the theoretical fundamentals of Fourier analysis.

Appendix D contains answers, hints, and solutions to the problems. This appendix is rather long, because in many cases it provides detailed solutions and sample program code. This should help students with little programming experience to develop the basic skills of using multiprocessor computer systems.

At the first reference in the book to the names of scientists and researchers, footnotes provide brief bibliographic information about them, taken from Wikipedia.

The notation list contains definitions of the most frequently used symbols.

The text is complemented with a detailed reference list.

Name and subject indexes conclude the book.

Thus, the availability in the book of both theoretical material and examples of solutions to computing problems, discussed in detail, as well as tasks for individual work, will make the book useful for both teaching parallel programming in higher education institutions and for use by researchers and programmers for computation on high-performance computer systems.

Voronezh, Russia — Sergei Kurgalin
Voronezh, Russia — Sergei Borzunov
July 2018

Contents

Notation

$\mathbb{N}$	The set of integers
$\mathbb{R}$	The set of real numbers
$\|A\|$	Cardinality of set A
$G(V, E)$	G is a graph with vertex set V and edge set E
$d(v)$	Degree of vertex v of a graph
$D(V, E)$	Digraph
$d^+(v)$	Outdegree of vertex v in a digraph
$d^-(v)$	Indegree of vertex v in a digraph
$\lfloor x \rfloor$	Floor function of x, i.e., the greatest integer less than or equal to the real number x
$\lceil x \rceil$	Ceiling function of x, i.e., the smallest integer greater than or equal to the real number x
A **and** B	Conjunction of logical expressions A and B
A **or** B	Disjunction of logical expressions A and B
$T_p(N)$	Time spent by a parallel algorithm executed on p processors on data of size N
$T_\infty(N)$	Time spent by a paracomputer on data of size N
$O(g(n))$	Class of functions growing not faster than $f(n)$
$\Omega(g(n))$	Class of functions growing at least as fast as $f(n)$
$\Theta(g(n))$	Class of functions growing at the same speed as $f(n)$
$S_p(N)$	Performance speedup using p processors on data of size N
$E_p(N)$	Efficiency of parallel computation using p processors on data of size N
$C_p(N)$	Cost of a parallel algorithm for p processors on data of size N
f	Proportion of code that is intrinsically sequential
$R(u)$	Set of input objects for statement u
$W(u)$	Set of output objects for statement u
$u \perp v$	Statements u and v are in a data dependency
$u \, \delta^0 \, v$	Data output dependency of statements u and v
$u \, \delta \, v$	Data flow dependency of statements u and v
$u \, \overline{\delta} \, v$	Data antidependency of statements u and v