Parallel Computers Architecture And Programming V Rajaraman

Programming Models for Parallel Computing
Parallel Computing
Algorithms, Software and Hardware of Parallel Computers
□□□□□□
Data-parallel Programming on MIMD Computers
Parallel Programming
Parallel Architectures and Their Efficient Use
Parallel Programming
Introduction to Parallel Processing
Raspberry Pi Super Cluster
Architectures, Algorithms, and Applications
Parallel Computing Works!
Architecture, Programming and Algorithms
Introduction to Parallel Computing
Architecture, Programming and Algorithms
A Hardware/software Approach
Principles of Parallel Computers and Some Impacts on Their Programming Models
A Hands-on Approach
Programming Massively Parallel Processors
Parallel Computers 2
Parallel Processing, 1980 to 2020
Models, Algorithms and Applications
Parallel Computer Architecture
PARALLEL COMPUTERS ARCHITECTURE AND PROGRAMMING
Practical Parallel Programming
From Algorithms to Programming on State-of-the-Art Platforms
for Multicore and Cluster Systems
for Multicore and Cluster Systems
Handbook of Parallel Computing and Statistics
First Heinz Nixdorf Symposium, Paderborn, Germany, November 11-13, 1992. Proceedings
Handbook of Parallel Computing
Scalable Parallel Computing
The Future of Computing Performance
□□□
Introduction to Parallel Computing
A Survey of Available Parallel Computer Systems
INTRODUCTION TO PARALLEL PROCESSING
Introduction to Parallel Computing

DARIO MICHAEL

Programming Models for Parallel Computing Springer Science & Business Media
Although multicore is now a mainstream architecture, there are few textbooks that cover parallel multicore architectures. Filling this gap, Fundamentals of Parallel Multicore Architecture provides all the material for a graduate or senior undergraduate course that focuses on the architecture of multicore processors. The book is also useful as a ref
Parallel Computing CRC Press
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 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, including 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 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.
CRC Press
Parallel computers have become widely available in recent years. Many scientists are now using them to investigate the grand challenges of science, such as modeling global climate change, determining the masses of elementary particles from first principles, or sequencing the human genome. However, software for parallel computers has developed far more slowly than the hardware. Many incompatible programming systems exist, and many useful programming techniques are not widely known. Practical Parallel Programming provides scientists and engineers with a detailed, informative, and often critical introduction to parallel programming techniques. Following a review of the fundamentals of parallel computer theory and architecture, it describes four of the most popular parallel programming models in use today—data parallelism, shared variables, message passing, and Linda—and shows how each can be used to solve various scientific and numerical problems. Examples, coded in various dialects of Fortran, are drawn from such domains as the solution of partial differential equations, solution of linear equations, the simulation of cellular automata, studies of rock fracturing, and image processing. Practical Parallel Programming will be particularly helpful for scientists and engineers who use high-performance computers to solve numerical problems and do physical simulations but who have little experience of networking or concurrency. The book can also be used by advanced undergraduate and graduate students in computer science in conjunction with material covering parallel architectures and algorithms in more detail. Computer science students will gain a critical appraisal of the current state of the art in parallel programming. Scientific and Engineering Computation series
Algorithms, Software and Hardware of Parallel Computers Morgan & Claypool Publishers
□□□□□□□□□□□□□□□□□□□□
□□□□□□ Pearson Education
Parallel Computing Architectures and APIs: IoT Big Data Stream Processing commences from the point high-performance uniprocessors were becoming increasingly complex, expensive, and power-hungry. A basic trade-off exists between the use of one or a small number of such complex processors, at one extreme, and a moderate to very large number of simpler processors, at the other. When combined with a high-bandwidth, interprocessor communication facility leads to significant simplification of the design process. However, two major roadblocks prevent the widespread adoption of such moderately to massively parallel architectures: the interprocessor communication bottleneck, and the difficulty and high cost of algorithm/software development. One of the most important reasons for studying parallel computing architectures is to learn how to extract the best performance from parallel systems. Specifically, you must understand its architectures so that you will be able to exploit those architectures during programming via the standardized APIs. This book would be useful for analysts, designers and developers of high-throughput computing systems essential for big data stream processing emanating from IoT-driven cyber-physical systems (CPS). This pragmatic book: Devolves uniprocessors in terms of a ladder of abstractions to ascertain (say) performance characteristics at a particular level of abstraction Explains limitations of uniprocessor high performance because of Moore’s Law Introduces basics of processors, networks and distributed systems Explains characteristics of parallel systems, parallel computing models and parallel algorithms Explains the three primary categorical representatives of parallel computing architectures, namely, shared memory, message passing and stream processing Introduces the three primary categorical representatives of parallel programming APIs, namely, OpenMP, MPI and CUDA Provides an overview of Internet of Things (IoT), wireless sensor networks (WSN), sensor data processing, Big Data and stream processing Provides introduction to 5G communications, Edge and Fog computing Parallel Computing Architectures and APIs: IoT Big Data Stream Processing discusses stream processing that enables the gathering, processing and analysis of high-volume, heterogeneous, continuous Internet of Things (IoT) big data streams, to extract insights and actionable results in real time. Application domains requiring data stream management include military, homeland security, sensor networks, financial applications, network management, web site performance tracking, real-time credit card fraud detection, etc.
Data-parallel Programming on MIMD Computers National Academies Press
An overview of the most prominent contemporary parallel processing programming models, written in a unique tutorial style. With the coming of the parallel computing era, computer scientists have turned their attention to designing programming models that are suited for high-performance parallel computing and supercomputing systems. Programming parallel systems is complicated by the fact that multiple processing units are simultaneously computing and moving data. This book offers an overview of some of the most prominent parallel programming models used in high-performance computing and supercomputing systems today. The chapters describe the programming models in a unique tutorial style rather than using the formal approach taken in the research literature. The aim is to cover a wide range of parallel programming models, enabling the reader to understand what each has to offer. The book begins with a description of the Message Passing Interface (MPI), the most common parallel programming model for distributed memory computing. It goes on to cover one-sided communication models, ranging from low-level runtime libraries (GASNet, OpenSHMEM) to high-level programming models (UPC, GA, Chapel); task-oriented programming models (Charm++, ADLB, Scioto, Swift, CnC) that allow users to describe their computation and data units as tasks so that the runtime system can manage computation and data movement as necessary; and parallel programming models intended for on-node parallelism in the context of multicore architecture or attached accelerators (OpenMP, Cilk Plus, TBB, CUDA, OpenCL). The book will be a valuable resource for graduate students, researchers, and any scientist who works with data sets and large computations. Contributors Timothy Armstrong, Michael G. Burke, Ralph Butler, Bradford L. Chamberlain, Sunita Chandrasekaran, Barbara Chapman, Jeff Daily, James Dinan, Deepak Eachempati, Ian T. Foster, William D. Gropp, Paul Hargrove, Wen-mei Hwu, Nikhil Jain, Laxmikant Kale, David Kirk, Kath Knobe, Ariram Krishnamoorthy, Jeffery A. Kuehn, Alexey Kukanov, Charles E. Leiserson, Jonathan Lifflander, Ewing Lusk, Tim Mattson, Bruce Palmer, Steven C. Pieper, Stephen W. Poole, Arch D. Robison, Frank Schlimbach, Rajeev Thakur, Abhinav Vishnu, Justin M. Wozniak, Michael Wilde, Kathy Yelick, Yili Zheng
Parallel Programming CRC Press

Advancements in microprocessor architecture, interconnection technology, and software development have fueled rapid growth in parallel and distributed computing. However, this development is only of practical benefit if it is accompanied by progress in the design, analysis and programming of parallel algorithms. This concise textbook provides, in one place, three mainstream parallelization approaches, Open MPP, MPI and OpenCL, for multicore computers, interconnected computers and graphical processing units. An overview of practical parallel computing and principles will enable the reader to design efficient parallel programs for solving various computational problems on state-of-the-art personal computers and computing clusters. Topics covered range from parallel algorithms, programming tools, OpenMP, MPI and OpenCL, followed by experimental measurements of parallel programs' run-times, and by engineering analysis of obtained results for improved parallel execution performances. Many examples and exercises support the exposition.
Parallel Architectures and Their Efficient Use MIT Press
PARALLEL COMPUTERS ARCHITECTURE AND PROGRAMMINGPHI Learning Pvt. Ltd.
Parallel Programming CRC Press
Past, Present, Parallel is a survey of the current state of the parallel processing industry. In the early 1980s, parallel computers were generally regarded as academic curiosities whose natural environment was the research laboratory. Today, parallelism is being used by every major computer manufacturer, although in very different ways, to produce increasingly powerful and cost-effec- tive machines. The first chapter introduces the basic concepts of parallel computing; the subsequent chapters cover different forms of parallelism, including descriptions of vector supercomputers, SIMD computers, shared memory multiprocessors, hypercubes, and transputer-based machines. Each section concentrates on a different manufacturer, detailing its history and company profile, the machines it currently produces, the software environments it supports, the market segment it is targetting, and its future plans. Supplementary chapters describe some of the companies which have been unsuccessful, and discuss a number of the common software systems which have been developed to make parallel computers more usable. The appendices describe the technologies which underpin parallelism. Past, Present, Parallel is an invaluable reference work, providing up-to-date material for commercial computer users and manufacturers, and for researchers and postgraduate students with an interest in parallel computing.
Introduction to Parallel Processing Springer Science & Business Media
The end of dramatic exponential growth in single-processor performance marks the end of the dominance of the single microprocessor in computing. The era of sequential computing must give way to a new era in which parallelism is at the forefront. Although important scientific and engineering challenges lie ahead, this is an opportune time for innovation in programming systems and computing architectures. We have already begun to see diversity in computer designs to optimize for such considerations as power and throughput. The next generation of discoveries is likely to require advances at both the hardware and software levels of computing systems. There is no guarantee that we can make parallel computing as common and easy to use as yesterday's sequential single-processor computer systems, but unless we aggressively pursue efforts suggested by the recommendations in this book, it will be "game over" for growth in computing performance. If parallel programming and related software efforts fail to become widespread, the development of exciting new applications that drive the computer industry will stall; if such innovation stalls, many other parts of the economy will follow suit. The Future of Computing Performance describes the factors that have led to the future limitations on growth for single processors that are based on complementary metal oxide semiconductor (CMOS) technology. It explores challenges inherent in parallel computing and architecture, including ever-increasing power consumption and the escalated requirements for heat dissipation. The book delineates a research, practice, and education agenda to help overcome these challenges. The Future of Computing Performance will guide researchers, manufacturers, and information technology professionals in the right direction for sustainable growth in computer performance, so that we may all enjoy the next level of benefits to society.
Raspberry Pi Super Cluster PHI Learning Pvt. Ltd.
Master efficient parallel programming to build powerful applications using Python About This Book Design and implement efficient parallel software Master new programming techniques to address and solve complex programming problems Explore the world of parallel programming with this book, which is a go-to resource for different kinds of parallel computing tasks in Python, using examples and topics covered in great depth Who This Book Is For Python Parallel Programming Cookbook is intended for software developers who are well versed with Python and want to use parallel programming techniques to write powerful and efficient code. This book will help you master the basics and the advanced of parallel computing. What You Will Learn Synchronize multiple threads and processes to manage parallel tasks Implement message passing communication between processes to build parallel applications Program your own GPU cards to address complex problems Manage computing entities to execute distributed computational tasks Write efficient programs by adopting the event-driven programming model Explore the cloud technology with DJango and Google App Engine Apply parallel programming techniques that can lead to performance improvements In Detail Parallel programming techniques are required for a developer to get the best use of all the computational resources available today and to build efficient software systems. From multi-core to GPU systems up to the distributed architectures, the high computation of programs throughout requires the use of programming tools and software libraries. Because of this, it is becoming increasingly important to know what the parallel programming techniques are. Python is commonly used as even non-experts can easily deal with its concepts. This book will teach you parallel programming techniques using examples in Python and will help you explore the many ways in which you can write code that allows more than one process to happen at once. Starting with introducing you to the world of parallel computing, it moves on to cover the fundamentals in Python. This is followed by exploring the thread-based parallelism model using the Python threading module by synchronizing threads and using locks, mutex, semaphores queues, GIL, and the thread pool. Next you will be taught about process-based parallelism where you will synchronize processes using message passing along with learning about the performance of MPI Python Modules. You will then go on to learn the asynchronous parallel programming model using the Python asyncio module along with handling exceptions. Moving on, you will discover distributed computing with Python, and learn how to install a broker, use Celery Python Module, and create a worker. You will also understand the StarCluster framework, Pycsp, Scoop, and Disco modules in Python. Further on, you will learn GPU programming with Python using the PyCUDA module along with evaluating performance limitations. Next you will get acquainted with the cloud computing concepts in Python, using Google App Engine (GAE), and building your first application with GAE. Lastly, you will learn about grid computing concepts in Python and using PyGlobus toolkit, GFTP and GASS COPY to transfer files, and service monitoring in PyGlobus. Style and approach A step-by-step guide to parallel programming using Python, with recipes accompanied by one or more programming examples. It is a practically oriented book and has all the necessary underlying parallel computing concepts.
Architectures, Algorithms, and Applications CRC Press
Research in the field of parallel computer architectures and parallel algorithms has been very successful in recent years, and further progress isto be expected. On the other hand, the question of basic principles of the architecture of universal parallel computers and their realizations is still wide open. The answer to this question must be regarded as mostimportant for the further development of parallel computing and especially for user acceptance. The First Heinz Nixdorf Symposium brought together leading experts in the field of parallel computing and its applications to discuss the state of the art, promising directions of research, and future perspectives. It was the first in a series of Heinz Nixdorf Symposia,intended to cover varying subjects from the research spectrum of the Heinz Nixdorf Institute of the University of Paderborn. This volume presents the proceedings of the symposium, which was held in Paderborn in November 1992. The contributions are grouped into four parts: parallel computation models and simulations, existing parallel machines, communication and programming paradigms, and parallel algorithms.
Parallel Computing Works! IOS Press
A clear illustration of how parallel computers can be successfully applied to large-scale scientific computations. This book demonstrates how a variety of applications in physics, biology, mathematics and other sciences were implemented on real parallel computers to produce new scientific results. It investigates issues of fine-grained parallelism relevant for future supercomputers with particular emphasis on hypercube architecture. The authors describe how they used an experimental approach to configure different massively parallel machines, design and implement basic system software, and develop algorithms for frequently used mathematical computations. They also devise performance models, measure the performance characteristics of several computers, and create a high-performance computing facility based exclusively on parallel computers. By addressing all issues involved in scientific problem solving, Parallel Computing Works! provides valuable insight into computational science for large-scale parallel architectures. For those in the sciences, the findings reveal the usefulness of an important experimental tool. Anyone in supercomputing and related computational fields will gain a new perspective on the potential contributions of parallelism. Includes over 30 full-color illustrations.
Architecture, Programming and Algorithms PARALLEL COMPUTERS ARCHITECTURE AND PROGRAMMING
THE CONTEXT OF PARALLEL PROCESSING The field of digital computer architecture has grown explosively in the past two decades. Through a steady stream of experimental research, tool-building efforts, and theoretical studies, the design of an instruction-set architecture, once considered an art, has been transformed into one of the most quantitative branches of computer technology. At the same time, better understanding of various forms of concurrency, from standard pipelining to massive parallelism, and invention of architectural structures to support a reasonably efficient and user-friendly programming model for such systems, has allowed hardware performance to continue its exponential growth. This trend is expected to continue in the near future. This explosive growth, linked with the expectation that performance will continue its exponential rise with each new generation of hardware and that (in stark contrast to software) computer hardware will function correctly as soon as it comes off the assembly line, has its down side. It has led to unprecedented hardware complexity and almost intolerable dev- opment costs. The challenge facing current and future computer designers is to institute simplicity where we now have complexity; to use fundamental theories being developed in this area to gain performance and ease-of-use benefits from simpler circuits; to understand the interplay between technological capabilities and limitations, on the one hand, and design decisions based on user and application requirements on the other.
Packt Publishing Ltd
Since the publication of the first edition, parallel computing technology has gained considerable momentum. A large proportion of this has come from the improvement in VLSI techniques, offering one to two orders of magnitude more devices than previously possible. A second contributing factor in the fast development of the subject is commercialization. The supercomputer is no longer restricted to a few well-established research institutions and large companies. A new computer breed combining the architectural advantages of the supercomputer with the advance of VLSI technology is now available at very attractive prices. A pioneering device in this development is the transputer, a VLSI processor specifically designed to operate in large concurrent systems. Parallel Computers 2: Architecture, Programming and Algorithms reflects the shift in emphasis of parallel computing and tracks the development of supercomputers in the years since the first edition was published. It looks at large-scale parallelism as found in transputer ensembles. This extensively rewritten second edition includes major new sections on the transputer and the OCCAM language. The book contains specific information on the various types of machines available, details of computer architecture and technologies, and descriptions of programming languages and algorithms. Aimed at an advanced undergraduate and postgraduate level, this handbook is also useful for research workers, machine designers, and programmers concerned with parallel computers. In addition, it will serve as a guide for potential parallel computer users, especially in disciplines where large amounts of computer time are regularly used.
Introduction to Parallel Computing Cambridge University Press
This book follows a step-by-step, tutorial-based approach which will teach you how to develop your own super cluster using Raspberry Pi computers quickly and efficiently. Raspberry Pi Super Cluster is an introductory guide for those interested in experimenting with parallel computing at home. Aimed at Raspberry Pi enthusiasts, this book is a primer for getting your first cluster up and running. Basic knowledge of C or Java would be helpful but no prior knowledge of parallel computing is necessary.
Architecture, Programming and Algorithms PHI Learning Pvt. Ltd.
Both algorithms and the software . and hardware of automatic computers have gone through a rapid development in the past 35 years. The dominant factor in this development was the advance in computer technology. Computer parameters were systematically improved through electron tubes, transistors and integrated circuits of ever-increasing integration density, which also influenced the development of new algorithms and programming methods. Some years ago the situation in computers development was that no additional enhancement of their performance could be achieved by increasing the speed of their logical elements, due to the physical barrier of the maximum transfer speed of electric signals. Another enhancement of computer performance has been achieved by parallelism, which makes it possible by a suitable organization of n processors to obtain a perform ance increase of up to n times. Research into parallel computations has been carried out for several years in many countries and many results of fundamental importance have been obtained. Many parallel computers have been designed and their algorithmic and program ming systems built. Such computers include ILLIAC IV, DAP, STARAN, OMEN, STAR-100, TEXAS INSTRUMENTS ASC, CRAY-1, C mmp, CM*, CLIP-3, PEPE. This trend is supported by the fact that: a) many algorithms and programs are highly parallel in their structure, b) the new LSI and VLSI technologies have allowed processors to be combined into large parallel structures, c) greater and greater demands for speed and reliability of computers are made.
A Hardware/software Approach Springer Science & Business Media
Today, parallel computing arouses enormous interest among students and professionals as it is clear that, as the new millennium progresses, all computers will work in parallel. A basic knowledge of the design and use of parallel computers is, therefore, essential for both students of computing and users of computers.Designed as an introductory-level textbook for the final year undergraduate students of computer science and engineering, this well-organized book covers state-of-the-art principles and techniques for designing and programming parallel computers. In the process,

Professor Rajaraman and Dr. Siva Ram Murthy, with their wealth of knowledge and years of teaching and research experience, give a masterly analysis of the various aspects of parallel computing.The book begins with an introduction to the current state and developments in parallel computing, then it goes on to give a detailed discussion on such topics as instruction level parallel processing, architecture of parallel computers, parallel algorithms and parallel programming. Besides, the book gives an in-depth coverage of compiler transformations and operating systems for parallel computers. The text concludes with a chapter on performance evaluation of parallel computers.Interspersed with copious examples and numerous exercises, this timely book should prove to be a handy and treasured volume for students as well as professionals.

Principles of Parallel Computers and Some Impacts on Their Programming Models Simon and Schuster

Today all computers, from tablet/desktop computers to super computers, work in parallel. A basic knowledge of the architecture of parallel computers and how to program them, is thus, essential for students of computer science and IT professionals. In its second edition, the book retains the lucidity of the first edition and has added new material to reflect the advances in parallel computers. It is designed as text for the final year undergraduate students of computer science and engineering and information technology. It describes the principles of designing parallel computers and how to program them. This second edition, while retaining the general structure of the earlier book, has added two new chapters, ‘Core Level Parallel Processing’ and ‘Grid and Cloud Computing’ based on the emergence of parallel computers on a single silicon chip popularly known as multicore processors and the rapid developments in Cloud Computing. All chapters have been revised and some chapters are re-written to reflect the emergence of multicore processors and the use of MapReduce in processing vast amounts of data. The new edition begins with an introduction to how to solve problems in parallel and describes how parallelism is used in improving the performance of computers. The topics discussed include instruction level parallel processing, architecture of parallel computers, multicore processors, grid and cloud computing, parallel algorithms, parallel programming, compiler transformations, operating systems for parallel computers, and performance evaluation of parallel computers.

A Hands-on Approach John Wiley & Sons

Mathematics of Computing -- Parallelism.