Parallel Computing For Data Science With Examples In R C And Cuda Chapman Hallcrc The R Series

Numerical algorithms, modern programming techniques, and parallel computing are often taught serially across different courses and different textbooks. The need to integrate concepts and tools usually comes only in employment or in research - after the courses are concluded - forcing the student to synthesise what is perceived to be three independent subfields into one. This book provides a seamless approach to stimulate the student simultaneously through the eyes of multiple disciplines, leading to enhanced understanding of scientific computing as a whole. The book includes both basic as well as advanced topics and places equal emphasis on the discretization of partial differential equations and on solvers. Some of the advanced topics include wavelets, high-order methods, non-symmetric systems, and parallelization of sparse systems. The material covered is suited to students from engineering, computer science, physics and mathematics.
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.
The fast and easy way to learn Python programming and statistics Python is a general-purpose programming language created in the late 1980s—and named after Monty Python—that's used by thousands of people to do things from testing microchips at Intel, to powering Instagram, to building video games with the PyGame library. Python For Data Science For Dummies is written for people who are new to data analysis, and discusses the basics of Python data analysis programming and statistics. The book also discusses Google Colab, which makes it possible to write Python code in the cloud. Get started with data science and Python Visualize information Wrangle data Learn from data The book provides the statistical background needed to get started in data science programming, including probability, random distributions, hypothesis testing, confidence intervals, and building regression models for prediction.
Mathematics of Computing -- Parallelism.
Deep Learning and Parallel Computing Environment for Bioengineering Systems delivers a significant forum for the technical advancement of deep learning in parallel computing environment across bio-engineering diversified domains and its applications. Pursuing an interdisciplinary approach, it focuses on methods used to identify and acquire valid, potentially useful knowledge sources. Managing the gathered knowledge and applying it to multiple domains including health care, social networks, mining, recommendation systems, image processing, pattern recognition and predictions using deep learning paradigms is the major strength of this book. This book integrates the core ideas of deep learning and its applications in bio engineering application domains, to be accessible to all scholars and academicians. The proposed techniques and concepts in this book can be extended in future to accommodate changing business organizations’ needs as well as practitioners’ innovative ideas. Presents novel, in-depth research contributions from a methodological/application perspective in understanding the fusion of deep machine learning paradigms and their capabilities in solving a diverse range of problems Illustrates the state-of-the-art and recent developments in the new theories and applications of deep learning approaches applied to parallel computing environment in bioengineering systems Provides concepts and technologies that are successfully used in the implementation of today's intelligent data-centric critical systems and multi-media Cloud-Big data
Parallel Computing for Data Science: With Examples in R, C++ and CUDA is one of the first parallel computing books to concentrate exclusively on parallel data structures, algorithms, software tools, and applications in data science. It includes examples not only from the classic "n observations, p variables" matrix format but also from time series,
This book discusses the current research and concepts in data science and how these can be addressed using different nature-inspired optimization techniques. Focusing on various data science problems, including classification, clustering, forecasting, and deep learning, it explores how researchers are using nature-inspired optimization techniques to find solutions to these problems in domains such as disease analysis and health care, object recognition, vehicular ad-hoc networking, high-dimensional data analysis, gene expression analysis, microgrids, and deep learning. As such it provides insights and inspiration for researchers to wanting to employ nature-inspired optimization techniques in their own endeavors.
A Tour of Data Science: Learn R and Python in Parallel covers the fundamentals of data science, including programming, statistics, optimization, and machine learning in a single short book. It does not cover everything, but rather, teaches the key concepts and topics in Data Science. It also covers two of the most popular programming languages used in Data Science, R and Python, in one source. Key features: Allows you to learn R and Python in parallel Cover statistics, programming, optimization and predictive modelling, and the popular data manipulation tools – data.table and pandas Provides a concise and accessible presentation Includes machine learning algorithms implemented from scratch, linear regression, lasso, ridge, logistic regression, gradient boosting trees, etc. Appealing to data scientists, statisticians, quantitative analysts, and others who want to learn programming with R and Python from a data science perspective.
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.

Parallel processing has been an enabling technology in scientific computing for more than 20 years. This book is the first in-depth discussion of parallel computing in 10 years; it reflects the mix of topics that mathematicians, computer scientists, and computational scientists focus on to make parallel processing effective for scientific problems. Presently, the impact of parallel processing on scientific computing varies greatly across disciplines, but it plays a vital role in most problem domains and is absolutely essential in many of them. Parallel Processing for Scientific Computing is divided into four parts: The first concerns performance modeling, analysis, and optimization; the second focuses on parallel algorithms and software for an array of problems common to many modeling and simulation applications; the third emphasizes tools and environments that can ease and enhance the process of application development; and the fourth provides a sampling of applications that require parallel computing for scaling to solve larger and realistic models that can advance science and engineering.

Big Data has been much in the news in recent years, and the advantages conferred by the collection and analysis of large datasets in fields such as marketing, medicine and finance have led to claims that almost any real world problem could be solved if sufficient data were available. This is of course a very simplistic view, and the usefulness of collecting, processing and storing large datasets must always be seen in terms of the communication, processing and storage capabilities of the computing platforms available. This book presents papers from the International Research Workshop, Advanced High Performance Computing Systems, held in Cetraro, Italy, in July 2014. The papers selected for publication here discuss fundamental aspects of the definition of Big Data, as well as considerations from practice where complex datasets are collected, processed and stored. The concepts, problems, methodologies and solutions presented are of much more general applicability than may be suggested by the particular application areas considered. As a result the book will be of interest to all those whose work involves the processing of very large data sets, exascale computing and the emerging fields of data science

Scientific computing has become an indispensable tool in numerous fields, such as physics, mechanics, biology, finance and industry. For example, it enables us, thanks to efficient algorithms adapted to current computers, to simulate, without the help of models or experimentations, the deflection of beams in bending, the sound level in a theater room or a fluid flowing around an aircraft wing. This book presents the scientific computing techniques applied to parallel computing for the numerical simulation of large-scale problems; these problems result from systems modeled by partial differential equations. Computing concepts will be tackled via examples. Implementation and programming techniques resulting from the finite element method will be presented for direct solvers, iterative solvers and domain decomposition methods, along with an introduction to MPI and OpenMP.

Master the robust features of R parallel programming to accelerate your data science computations About This Book Create R programs that exploit the computational capability of your cloud platforms and computers to the fullest Become an expert in writing the most efficient and highest performance parallel algorithms in R Get to grips with the concept of parallelism to accelerate your existing R programs Who This Book Is For This book is for R programmers who want to step beyond its inherent single-threaded and restricted memory limitations and learn how to implement highly accelerated and scalable algorithms that are a necessity for the performant processing of Big Data. No previous knowledge of parallelism is required. This book also provides for the more advanced technical programmer seeking to go beyond high level parallel frameworks. What You Will Learn Create and structure efficient load-balanced parallel computation in R, using R's built-in parallel package Deploy and utilize cloud-based parallel infrastructure from R, including launching a distributed computation on Hadoop running on Amazon Web Services (AWS) Get accustomed to parallel efficiency, and apply simple techniques to benchmark, measure speed and target improvement in your own code Develop complex parallel processing algorithms with the standard Message Passing Interface (MPI) using RMPI, pbdMPI, and SPRINT packages Build and extend a parallel R package (SPRINT) with your own MPI-based routines Implement accelerated numerical functions in R utilizing the vector processing capability of your Graphics Processing Unit (GPU) with OpenCL Understand parallel programming pitfalls, such as deadlock and numerical instability, and the approaches to handle and avoid them Build a task farm master-worker, spatial grid, and hybrid parallel R programs In Detail R is one of the most popular programming languages used in data science. Applying R to big data and complex analytic tasks requires the harnessing of scalable compute resources. Mastering Parallel Programming with R presents a comprehensive and practical treatise on how to build highly scalable and efficient algorithms in R. It will teach you a variety of parallelization techniques, from simple use of R's built-in parallel package versions of lapply(), to high-level AWS cloud-based Hadoop and Apache Spark frameworks. It will also teach you low level scalable parallel programming using RMPI and pbdMPI for message passing, applicable to clusters and supercomputers, and how to exploit thousand-fold simple processor GPUs through ROpenCL. By the end of the book, you will understand the factors that influence parallel efficiency, including assessing code performance and implementing load balancing; pitfalls to avoid, including deadlock and numerical instability issues; how to structure your code and data for the most appropriate type of parallelism for your problem domain; and how to extract the maximum performance from your R code running on a variety of computer systems. Style and approach This book leads you chapter by chapter from the easy to more complex forms of parallelism. The author's insights are presented through clear practical examples applied to a range of different problems, with comprehensive reference information for each of the R packages employed. The book can be read from start to finish, or by dipping in chapter by chapter, as each chapter describes a specific parallel approach and technology, so can be read as a standalone.

Foreword by Bjarne Stroustrup Software is generally acknowledged to be the single greatest obstacle preventing mainstream adoption of massively-parallel computing. While sequential applications are routinely ported to platforms ranging from PCs to mainframes, most parallel programs only ever run on one type of machine. One reason for this is that most parallel programming systems have failed to insulate their users from the architectures of the machines on which they have run. Those that have been platform-independent have usually also had poor performance. Many researchers now believe that object-oriented languages may offer a solution. By hiding the architecture-specific constructs required for high performance inside platform-independent abstractions, parallel object-oriented programming systems may be able to combine the speed of massively-parallel computing with the comfort of sequential programming. Parallel Programming Using C++ describes fifteen parallel programming systems

based on C++, the most popular object-oriented language of today. These systems cover the whole spectrum of parallel programming paradigms, from data parallelism through dataflow and distributed shared memory to message-passing control parallelism. For the parallel programming community, a common parallel application is discussed in each chapter, as part of the description of the system itself. By comparing the implementations of the polygon overlay problem in each system, the reader can get a better sense of their expressiveness and functionality for a common problem. For the systems community, the chapters contain a discussion of the implementation of the various compilers and runtime systems. In addition to discussing the performance of polygon overlay, several of the contributors also discuss the performance of other, more substantial, applications. For the research community, the contributors discuss the motivations for and philosophy of their systems. As well, many of the chapters include critiques that complete the research arc by pointing out possible future research directions. Finally, for the object-oriented community, there are many examples of how encapsulation, inheritance, and polymorphism can be used to control the complexity of developing, debugging, and tuning parallel software.

This edited book aims to present the state of the art in research and development of the convergence of high-performance computing and parallel programming for various engineering and scientific applications. The book has consolidated algorithms, techniques, and methodologies to bridge the gap between the theoretical foundations of academia and implementation for research, which might be used in business and other real-time applications in the future.The book outlines techniques and tools used for emergent areas and domains, which include acceleration of large-scale electronic structure simulations with heterogeneous parallel computing, characterizing power and energy efficiency of a data-centric high-performance computing runtime and applications, security applications of GPUs, parallel implementation of multiprocessors on MPI using FDTD, particle-based fused rendering, design and implementation of particle systems for mesh-free methods with high performance, and evolving topics on heterogeneous computing. In the coming days the need to converge HPC, IoT, cloud-based applications will be felt and this volume tries to bridge that gap.

What does Google's management of billions of Web pages have in common with analysis of a genome with billions of nucleotides? Both apply methods that coordinate many processors to accomplish a single task. From mining genomes to the World Wide Web, from modeling financial markets to global weather patterns, parallel computing enables computations that would otherwise be impractical if not impossible with sequential approaches alone. Its fundamental role as an enabler of simulations and data analysis continues an advance in a wide range of application areas. Scientific Parallel Computing is the first textbook to integrate all the fundamentals of parallel computing in a single volume while also providing a basis for a deeper understanding of the subject. Designed for graduate and advanced undergraduate courses in the sciences and in engineering, computer science, and mathematics, it focuses on the three key areas of algorithms, architecture, languages, and their crucial synthesis in performance. The book's computational examples, whose math prerequisites are not beyond the level of advanced calculus, derive from a breadth of topics in scientific and engineering simulation and data analysis. The programming exercises presented early in the book are designed to bring students up to speed quickly, while the book later develops projects challenging enough to guide students toward research questions in the field. The new paradigm of cluster computing is fully addressed. A supporting web site provides access to all the codes and software mentioned in the book, and offers topical information on popular parallel computing systems. Integrates all the fundamentals of parallel computing essential for today's high-performance requirements Ideal for graduate and advanced undergraduate students in the sciences and in engineering, computer science, and mathematics Extensive programming and theoretical exercises enable students to write parallel codes quickly More challenging projects later in the book introduce research questions New paradigm of cluster computing fully addressed Supporting web site provides access to all the codes and software mentioned in the book

Parallel and High Performance Computing offers techniques guaranteed to boost your code’s effectiveness. Summary Complex calculations, like training deep learning models or running large-scale simulations, can take an extremely long time. Efficient parallel programming can save hours—or even days—of computing time. Parallel and High Performance Computing shows you how to deliver faster run-times, greater scalability, and increased energy efficiency to your programs by mastering parallel techniques for multicore processor and GPU hardware. About the technology Write fast, powerful, energy efficient programs that scale to tackle huge volumes of data. Using parallel programming, your code spreads data processing tasks across multiple CPUs for radically better performance. With a little help, you can create software that maximizes both speed and efficiency. About the book Parallel and High Performance Computing offers techniques guaranteed to boost your code’s effectiveness. You’ll learn to evaluate hardware architectures and work with industry standard tools such as OpenMP and MPI. You’ll master the data structures and algorithms best suited for high performance computing and learn techniques that save energy on handheld devices. You’ll even run a massive tsunami simulation across a bank of GPUs. What's inside Planning a new parallel project Understanding differences in CPU and GPU architecture Addressing underperforming kernels and loops Managing applications with batch scheduling About the reader For experienced programmers proficient with a high-performance computing language like C, C++, or Fortran. About the author Robert Robey works at Los Alamos National Laboratory and has been active in the field of parallel computing for over 30 years. Yuliana Zamora is currently a PhD student and Siebel Scholar at the University of Chicago, and has lectured on programming modern hardware at numerous national conferences. Table of Contents PART 1 INTRODUCTION TO PARALLEL COMPUTING 1 Why parallel computing? 2 Planning for parallelization 3 Performance limits and profiling 4 Data design and performance models 5 Parallel algorithms and patterns PART 2 CPU: THE PARALLEL WORKHORSE 6 Vectorization: FLOPs for free 7 OpenMP that performs 8 MPI: The parallel backbone PART 3 GPUS: BUILT TO ACCELERATE 9 GPU architectures and concepts 10 GPU programming model 11 Directive-based GPU programming 12 GPU languages: Getting down to basics 13 GPU profiling and tools PART 4 HIGH PERFORMANCE COMPUTING ECOSYSTEMS 14 Affinity: Truce with the kernel 15 Batch schedulers: Bringing order to chaos 16 File operations for a parallel world 17 Tools and resources for better code

There is a software gap between the hardware potential and the performance that can be attained using today's software parallel program development tools. The tools need manual intervention by the programmer to parallelize the code.

Programming a parallel computer requires closely studying the target algorithm or application, more so than in the traditional sequential programming we have all learned. The programmer must be aware of the communication and data dependencies of the algorithm or application. This book provides the techniques to explore the possible ways to program a parallel computer for a given application.

Get to grips with processing large volumes of data and presenting it as engaging, interactive insights using Spark and Python. Key Features Get a hands-on, fast-paced introduction to the Python data science stack Explore ways to create useful metrics and statistics from large datasets Create detailed analysis reports with real-world data Book Description Processing big data in real time is challenging due to scalability, information inconsistency, and fault tolerance. Big Data Analysis with Python teaches you how to use tools that can control this data avalanche for you. With this book, you'll learn practical techniques to aggregate data into useful dimensions for posterior analysis, extract statistical measurements, and transform datasets into features for other systems. The book begins with an introduction to data manipulation in Python using pandas. You'll then get familiar with statistical analysis and plotting techniques. With multiple hands-on activities in store, you'll be able to analyze data that is distributed on several computers by using Dask. As you progress, you'll study how to aggregate data for plots when the entire data cannot be accommodated in memory. You'll also explore Hadoop (HDFS and YARN), which will help you tackle larger datasets. The book also covers Spark and explains how it interacts with other tools. By the end of this book, you'll be able to bootstrap your own Python environment, process large files, and manipulate data to generate statistics, metrics, and graphs. What you will learn Use Python to read and transform data into different formats Generate basic statistics and metrics using data on disk Work with computing tasks distributed over a cluster Convert data from various sources into storage or querying formats Prepare data for statistical analysis, visualization, and machine learning Present data in the form of effective visuals Who this book is for Big Data Analysis with Python is designed for Python developers, data analysts, and data scientists who want to get hands-on with methods to control data and transform it into impactful insights. Basic knowledge of statistical measurements and relational databases will help you to understand various concepts explained in this book.

For the last four decades, parallel computing platforms have increasingly formed the basis for the development of high performance systems primarily aimed at the solution of intensive computing problems, and the application of parallel computing systems has also become a major factor in furthering scientific research. But such systems also offer the possibility of solving the problems encountered in the processing of large-scale scientific data sets, as well as in the analysis of Big Data in the fields of medicine, social media, marketing, economics etc. This book presents papers from the International Research Workshop on Advanced High Performance Computing Systems, held in Cetraro, Italy, in July 2016. The workshop covered a wide range of topics and new developments related to the solution of intensive and large-scale computing problems, and the contributions included in this volume cover aspects of the evolution of parallel platforms and highlight some of the problems encountered with the development of ever more powerful computing systems. The importance of future large-scale data science applications is also discussed. The book will be of particular interest to all those involved in the development or application of parallel computing systems.

Programming Massively Parallel Processors: A Hands-on Approach, Second Edition, teaches students how to program massively parallel processors. It offers a detailed discussion of various techniques for constructing parallel programs. Case studies are used to demonstrate the development process, which begins with computational thinking and ends with effective and efficient parallel programs. This guide shows both student and professional alike the basic concepts of parallel programming and GPU architecture. Topics of performance, floating-point format, parallel patterns, and dynamic parallelism are covered in depth. This revised edition contains more parallel programming examples, commonly-used libraries such as Thrust, and explanations of the latest tools. It also provides new coverage of CUDA 5.0, improved performance, enhanced development tools, increased hardware support, and more; increased coverage of related technology, OpenCL and new material on algorithm patterns, GPU clusters, host programming, and data parallelism; and two new case studies (on MRI reconstruction and molecular visualization) that explore the latest applications of CUDA and GPUs for scientific research and high-performance computing. This book should be a valuable resource for advanced students, software engineers, programmers, and hardware engineers. New coverage of CUDA 5.0, improved performance, enhanced development tools, increased hardware support, and more Increased coverage of related technology, OpenCL and new material on algorithm patterns, GPU clusters, host programming, and data parallelism Two new case studies (on MRI reconstruction and molecular visualization) explore the latest applications of CUDA and GPUs for scientific research and high-performance computing

The ability of parallel computing to process large data sets and handle time-consuming operations has resulted in unprecedented advances in biological and scientific computing, modeling, and simulations. Exploring these recent developments, the Handbook of Parallel Computing: Models, Algorithms, and Applications provides comprehensive coverage on a

It’s tough to argue with R as a high-quality, cross-platform, open source statistical software product—unless you’re in the business of crunching Big Data. This concise book introduces you to several strategies for using R to analyze large datasets, including three chapters on using R and Hadoop together. You’ll learn the basics of Snow, Multicore, Parallel, Segue, RHIPE, and Hadoop Streaming, including how to find them, how to use them, when they work well, and when they don’t. With these packages, you can overcome R’s single-threaded nature by spreading work across multiple CPUs, or offloading work to multiple machines to address R’s memory barrier. Snow: works well in a traditional cluster environment Multicore: popular for multiprocessor and multicore computers Parallel: part of the upcoming R 2.14.0 release R+Hadoop: provides low-level access to a popular form of cluster computing RHIPE: uses Hadoop’s power with R’s language and interactive shell Segue: lets you use Elastic MapReduce as a backend for lapply-style operations

Technological improvements continue to push back the frontier of processor speed in modern computers. Unfortunately, the computational intensity demanded by modern research problems grows even faster. Parallel computing has emerged as the most successful bridge to this computational gap, and many popular solutions have emerged based on its concepts

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

Data science has taken the world by storm. Every field of study and area of business has been affected as people increasingly realize the value of the incredible quantities of data being generated. But to extract value from those data, one needs to be tra

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.

This volume gives an overview of the state-of-the-art with respect to the development of all types of parallel computers and their application to a wide range of problem areas. The international conference on parallel computing ParCo97 (Parallel Computing 97) was held in Bonn, Germany from 19 to 22 September 1997. The first conference in this biannual series was held in 1983 in Berlin. Further conferences were held in Leiden (The Netherlands), London (UK), Grenoble (France) and Gent (Belgium). From the outset the aim with the ParCo (Parallel Computing) conferences was to promote the application of parallel computers to solve real life problems. In the case of ParCo97 a new milestone was reached in that more than half of the papers and posters presented were concerned with application aspects. This fact reflects the coming of age of parallel computing. Some 200 papers were submitted to the Program Committee by authors from all over the world. The final programme consisted of four invited papers, 71 contributed scientific/industrial papers and 45 posters. In addition a panel discussion on Parallel Computing and the Evolution of Cyberspace was held. During and after the conference all final contributions were refereed. Only those papers and posters accepted during this final screening process are included in this volume. The practical emphasis of the conference was accentuated by an industrial exhibition where companies demonstrated the newest developments in parallel processing equipment and software. Speakers from participating companies presented papers in industrial sessions in which new developments in parallel computing were reported.

Amazon's #1 Bestseller Dr. Ganapathi Pulipaka is a Chief Data Scientist and SAP Technical Lead for one of the largest firms in the world. He is also a PostDoc Research Scholar in Computer Science Engineering in Big Data Analytics, Machine Learning, Robotics, IoT, Artificial Intelligence as part of Doctor of Computer Science program from Colorado Technical University, CO with another PhD in Data Analytics, Information Systems, and Enterprise Resource Management, California University, Irvine. A technology leader in artificial intelligence, SAP development, and solution architecture. A project/program manager for application development of SAP systems, machine learning, deep learning systems, application development management, basis, infrastructure, and consulting delivery services offering expertise in delivery execution and executive interaction. Experienced implementing ASAP, Agile, Agile ASAP 8, HANA ASAP 8, Activate, Prince2, SCRUM, and Waterfall SDLC project methodologies. Implemented multiple SAP programs/projects managing a team size of 60+ members, managing budget more than $5M to $10M with SAP backend databases of Oracle, IBM DB2, Sybase, Informix, MS SQL server on Mac OS and Linux environments. His background is in Computer Science with a professional skillset and two decades of management and hands-on development experience in Machine Learning in TensorFlow, Python, and R, Deep Learning in TensorFlow, Python, and R, SAP ABAP S/4 HANA 1609, SAP S/4 H HANA 1710, SAP IBP on SAP Cloud Platform, Big Data, IaaS, IoT, Data Science, Apache Hadoop, Apache Kafka, Apache Spark, Apache Storm, Apache Flink, SQL, NoSQL, Tableau, PowerBI, Mathematics, Data Mining, Statistical Framework, SIEM, SAP, SAP ERP/ECC 6.0 NetWeaver Portals, SAP PLM, cProjects, R/3, BW, SRM 5.0, CRM 7.4, 7.3, 7.2, 7.1, 7.0, Java, C, C++, VC++, SAP CRM-IPM, SAP CRM- Service management, SAP CRM-Banking, SAP PLM Web UI 7.47, xRPM, SCM 7.1 APO, DP, SNP, SNC, FSCM, FSCD, SCEM, EDI. CRM ABAP/OO, ABAP, CRM Web UI/BOL/GENIL/ABAP Objects, SAP Netweaver Gateway (OData), SAP Mobility, SAP Fiori, Information Security, CyberSecurity, Governance, Risk Controls, and Compliance, SAP Fiori HANA, ABAP Webdynpros, BSPs, EDI/ALE, CRM Middleware, CRM Workflow, JavaScript, SAP KW 7.3 SAP Content server, SAP TREX Server, SAP KPro, SAP PI (PO), SAP BPC, Script logics, Azure, SAP BPM, SAP UI5, SAP BRM, Unix, Linux, macOS, and always looking for patterns in data and performing extractions to provide new meanings and insights through algorithms and analytics.

The book 'Data Intensive Computing Applications for Big Data' discusses the technical concepts of big data, data intensive computing through machine learning, soft computing and parallel computing paradigms. It brings together researchers to report their latest results or progress in the development of the above mentioned areas. Since there are few books on this specific subject, the editors aim to provide a common platform for researchers working in this area to exhibit their novel findings. The book is intended as a reference work for advanced undergraduates and graduate students, as well as multidisciplinary, interdisciplinary and transdisciplinary research workers and scientists on the subjects of big data and cloud/parallel and distributed computing, and explains didactically many of the core concepts of these approaches for practical applications. It is organized into 24 chapters providing a comprehensive overview of big data analysis using parallel computing and addresses the complete data science workflow in the cloud, as well as dealing with privacy issues and the challenges faced in

a data-intensive cloud computing environment. The book explores both fundamental and high-level concepts, and will serve as a manual for those in the industry, while also helping beginners to understand the basic and advanced aspects of big data and cloud computing.
Thought-provoking and accessible in approach, this updated and expanded second edition of the Parallel Computing for Data Science: With Examples in R, C++ and CUDA provides a user-friendly introduction to the subject, Taking a clear structural framework, it guides the reader through the subject's core elements. A flowing writing style combines with the use of illustrations and diagrams throughout the text to ensure the reader understands even the most complex of concepts. This succinct and enlightening overview is a required reading for advanced graduate-level students. We hope you find this book useful in shaping your future career. Feel free to send us your enquiries related to our publications to info@risepress.pw Rise Press
This highly acclaimed work, first published by Prentice Hall in 1989, is a comprehensive and theoretically sound treatment of parallel and distributed numerical methods. It focuses on algorithms that are naturally suited for massive parallelization, and it explores the fundamental convergence, rate of convergence, communication, and synchronization issues associated with such algorithms. This is an extensive book, which aside from its focus on parallel and distributed algorithms, contains a wealth of material on a broad variety of computation and optimization topics. It is an excellent supplement to several of our other books, including Convex Optimization Algorithms (Athena Scientific, 2015), Nonlinear Programming (Athena Scientific, 1999), Dynamic Programming and Optimal Control (Athena Scientific, 2012), Neuro-Dynamic Programming (Athena Scientific, 1996), and Network Optimization (Athena Scientific, 1998). The on-line edition of the book contains a 95-page solutions manual.
Summary Dask is a native parallel analytics tool designed to integrate seamlessly with the libraries you're already using, including Pandas, NumPy, and Scikit-Learn. With Dask you can crunch and work with huge datasets, using the tools you already have. And Data Science with Python and Dask is your guide to using Dask for your data projects without changing the way you work! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. You'll find registration instructions inside the print book. About the Technology An efficient data pipeline means everything for the success of a data science project. Dask is a flexible library for parallel computing in Python that makes it easy to build intuitive workflows for ingesting and analyzing large, distributed datasets. Dask provides dynamic task scheduling and parallel collections that extend the functionality of NumPy, Pandas, and Scikit-learn, enabling users to scale their code from a single laptop to a cluster of hundreds of machines with ease. About the Book Data Science with Python and Dask teaches you to build scalable projects that can handle massive datasets. After meeting the Dask framework, you'll analyze data in the NYC Parking Ticket database and use DataFrames to streamline your process. Then, you'll create machine learning models using Dask-ML, build interactive visualizations, and build clusters using AWS and Docker. What's inside Working with large, structured and unstructured datasets Visualization with Seaborn and Datashader Implementing your own algorithms Building distributed apps with Dask Distributed Packaging and deploying Dask apps About the Reader For data scientists and developers with experience using Python and the PyData stack. About the Author Jesse Daniel is an experienced Python developer. He taught Python for Data Science at the University of Denver and leads a team of data scientists at a Denver-based media technology company. Table of Contents PART 1 - The Building Blocks of scalable computing Why scalable computing matters Introducing Dask PART 2 - Working with Structured Data using Dask DataFrames Introducing Dask DataFrames Loading data into DataFrames Cleaning and transforming DataFrames Summarizing and analyzing DataFrames Visualizing DataFrames with Seaborn Visualizing location data with Datashader PART 3 - Extending and deploying Dask Working with Bags and Arrays Machine learning with Dask-ML Scaling and deploying Dask
Building upon the wide-ranging success of the first edition, Parallel Scientific Computation presents a single unified approach to using a range of parallel computers, from a small desktop computer to a massively parallel computer. The author explains how to use the bulk synchronous parallel (BSP) model to design and implement parallel algorithms in the areas of scientific computing and big data, and provides a full treatment of core problems in these areas, starting from a high-level problem description, via a sequential solution algorithm to a parallel solution algorithm and an actual parallel program written in BSPlib. Every chapter of the book contains a theoretical section and a practical section presenting a parallel program and numerical experiments on a modern parallel computer to put the theoretical predictions and cost analysis to the test. Every chapter also presents extensive bibliographical notes with additional discussions and pointers to relevant literature, and numerous exercises which are suitable as graduate student projects. The second edition provides new material relevant for big-data science such as sorting and graph algorithms, and it provides a BSP approach towards new hardware developments such as hierarchical architectures with both shared and distributed memory. A single, simple hybrid BSP system suffices to handle both types of parallelism efficiently, and there is no need to master two systems, as often happens in alternative approaches. Furthermore, the second edition brings all algorithms used up to date, and it includes new material on high-performance linear system solving by LU decomposition, and improved data partitioning for sparse matrix computations. The book is accompanied by a software package BSPedupack, freely available online from the author's homepage, which contains all programs of the book and a set of test driver programs. This package written in C can be run using modern BSPlib implementations such as MulticoreBSP for C or BSPonMPI.
A complete source of information on almost all aspects of parallel computing from introduction, to architectures, to programming paradigms, to algorithms, to programming standards. It covers traditional Computer Science algorithms, scientific computing algorithms and data intensive algorithms.
Parallel Computing for Data ScienceWith Examples in R, C++ and CUDACRC Press
This book constitutes the refereed proceedings of the International Conference on Soft Computing in Data Science, SCDS 2016, held in Putrajaya, Malaysia, in September 2016. The 27 revised full papers presented were carefully reviewed and selected from 66 submissions. The papers are organized in topical sections on artificial neural networks; classification, clustering, visualization; fuzzy logic; information and sentiment analytics.
This gentle introduction to High Performance Computing (HPC) for Data Science using the Message Passing Interface (MPI) standard has been designed as a first course for undergraduates on parallel programming on distributed memory models, and requires only basic programming notions. Divided into two parts the first part covers high performance computing using C++ with the Message Passing Interface (MPI) standard followed by a second part providing high-performance data analytics on computer clusters. In the first part, the fundamental notions of blocking versus non-blocking point-to-point communications, global communications (like broadcast or scatter) and collaborative computations (reduce), with Amdalh and Gustafson speed-up laws are described before addressing parallel sorting and parallel linear algebra on computer clusters. The common ring, torus and hypercube topologies of clusters are then explained and global communication procedures on these topologies are studied. This

first part closes with the MapReduce (MR) model of computation well-suited to processing big data using the MPI framework. In the second part, the book focuses on high-performance data analytics. Flat and hierarchical clustering algorithms are introduced for data exploration along with how to program these algorithms on computer clusters, followed by machine learning classification, and an introduction to graph analytics. This part closes with a concise introduction to data core-sets that let big data problems be amenable to tiny data problems. Exercises are included at the end of each chapter in order for students to practice the concepts learned, and a final section contains an overall exam which allows them to evaluate how well they have assimilated the material covered in the book.

In the twenty-first century, scientists and engineers are tackling the world's toughest computational problems with parallel computing. Using multiple processor cores running simultaneously, parallel computers are solving these problems in less time and with greater accuracy than ever before. Even desktop PCs nowadays are powerful parallel computers. To take full advantage of the capabilities of these machines, programmers must learn to write parallel programs. BIG CPU, BIG DATA teaches you how to write parallel programs for multicore machines, compute clusters, GPU accelerators, and big data map-reduce jobs, in the Java language, with the free, easy-to-use, object-oriented Parallel Java 2 Library. The book also covers how to measure the performance of parallel programs and how to design the programs to run as fast as possible.

Parallel computing has been the enabling technology of high-end machines for many years. Now, it has finally become the ubiquitous key to the efficient use of any kind of multi-processor computer architecture, from smart phones, tablets, embedded systems and cloud computing up to exascale computers. _x000D_ This book presents the proceedings of ParCo2013 – the latest edition of the biennial International Conference on Parallel Computing – held from 10 to 13 September 2013, in Garching, Germany. The conference focused on several key parallel computing areas. Themes included parallel programming models for multi- and manycore CPUs, GPUs, FPGAs and heterogeneous platforms, the performance engineering processes that must be adapted to efficiently use these new and innovative platforms, novel numerical algorithms and approaches to large-scale simulations of problems in science and engineering._x000D_ The conference programme also included twelve mini-symposia (including an industry session and a special PhD Symposium), which comprehensively represented and intensified the discussion of current hot topics in high performance and parallel computing. These special sessions covered large-scale supercomputing, novel challenges arising from parallel architectures (multi-/manycore, heterogeneous platforms, FPGAs), multi-level algorithms as well as multi-scale, multi-physics and multi-dimensional problems._x000D_ It is clear that parallel computing – including the processing of large data sets (“Big Data”) – will remain a persistent driver of research in all fields of innovative computing, which makes this book relevant to all those with an interest in this field.