Describing state-of-the-art solutions in distributed system architectures, Integration of Services into Workflow Applications presents a concise approach to the integration of loosely coupled services into workflow applications. It discusses key challenges related to the integration of distributed systems and proposes solutions, both in terms of theoretical aspects such as models and workflow scheduling algorithms, and technical solutions such as software tools and APIs. The book provides an in-depth look at workflow scheduling and proposes a way to integrate several different types of services into one single workflow application. It shows how these components can be expressed as services that can subsequently be integrated into workflow applications. The workflow applications are often described as acyclic graphs with dependencies which allow readers to define complex scenarios in terms of basic tasks. Presents state-of-the-art solutions to challenges in multi-domain workflow application definition, optimization, and execution. Proposes a uniform concept of a service that can represent executable components in all major distributed software architectures used today. Discusses an extended model with determination of data flows among parallel paths of a workflow application. Since workflow applications often process big data, the book explores the dynamic management of data with various storage constraints during workflow execution. It addresses several practical problems related to data handling, including data partitioning for parallel processing next to service selection and scheduling, processing data in batches or streams, and constraints on data sizes that can be processed at the same time by service instances. Illustrating several workflow applications that were proposed, implemented, and benchmarked in a real BeesyCluster environment, the
book includes templates for multidisciplinary workflow applications that readers can use in a wide range of contexts.

"OpenCL in Action blends the theory of parallel computing with the practical reality of building high-performance applications using OpenCL. It first guides you through the fundamental data structures in an intuitive manner. Then, it explains techniques for high-speed sorting, image processing, matrix operations, and fast Fourier transform. The book concludes with a deep look at the all-important subject of graphics acceleration. Numerous challenging examples give you different ways to experiment with working code."--Pub. desc.

This book follows an example-driven, simplified, and practical approach to using OpenCL for general purpose GPU programming. If you are a beginner in parallel programming and would like to quickly accelerate your algorithms using OpenCL, this book is perfect for you! You will find the diverse topics and case studies in this book interesting and informative. You will only require a good knowledge of C programming for this book, and an understanding of parallel implementations will be useful, but not necessary.

Cryptography, the science of encoding and decoding information, allows people to do online banking, online trading, and make online purchases, without worrying that their personal information is being compromised. The dramatic increase of information transmitted electronically has led to an increased reliance on cryptography. This book discusses the theories and concepts behind modern cryptography and demonstrates how to develop and implement cryptographic algorithms using C++ programming language. Written for programmers and engineers, Practical Cryptography explains how you can use cryptography to maintain the privacy of computer data. It describes dozens of cryptography algorithms, gives practical advice on how to implement them into cryptographic software, and shows how they can be used to solve security problems. Covering the latest developments in practical cryptographic techniques, this book shows you how to build security into your computer applications, networks, and storage. Suitable for undergraduate and postgraduate students in cryptography, network security, and other security-related courses, this book will also help anyone involved in computer and network security who wants to learn the nuts and bolts of practical cryptography.

This book makes powerful Field Programmable Gate Array (FPGA) and reconfigurable technology accessible to software engineers by covering different state-of-the-art high-level synthesis approaches (e.g., OpenCL and several C-to-gates compilers). It introduces FPGA technology, its programming model, and how various applications can be implemented on FPGAs without going through low-level hardware design phases. Readers will get a realistic sense for problems that are suited for FPGAs and how to implement them from a software designer’s point of view. The authors demonstrate that FPGAs and their programming model reflect the needs of stream processing problems much better than traditional CPU or GPU architectures, making them well-suited for a wide variety of systems, from embedded systems performing sensor processing to large setups for Big Data number crunching. This book serves as an invaluable tool for software designers and FPGA design engineers who are interested in high design
productivity through behavioural synthesis, domain-specific compilation, and FPGA overlays. Introduces FPGA technology to software developers by giving an overview of FPGA programming models and design tools, as well as various application examples; Provides a holistic analysis of the topic and enables developers to tackle the architectural needs for Big Data processing with FPGAs; Explains the reasons for the energy efficiency and performance benefits of FPGA processing; Provides a user-oriented approach and a sense for where and how to apply FPGA technology.

This four volume set LNCS 9528, 9529, 9530 and 9531 constitutes the refereed proceedings of the 15th International Conference on Algorithms and Architectures for Parallel Processing, ICA3PP 2015, held in Zhangjiajie, China, in November 2015. The 219 revised full papers presented together with 77 workshop papers in these four volumes were carefully reviewed and selected from 807 submissions (602 full papers and 205 workshop papers). The first volume comprises the following topics: parallel and distributed architectures; distributed and network-based computing and internet of things and cyber-physical-social computing. The second volume comprises topics such as big data and its applications and parallel and distributed algorithms. The topics of the third volume are: applications of parallel and distributed computing and service dependability and security in distributed and parallel systems. The covered topics of the fourth volume are: software systems and programming models and performance modeling and evaluation.

In view of the growing presence and popularity of multicore and manycore processors, accelerators, and coprocessors, as well as clusters using such computing devices, the development of efficient parallel applications has become a key challenge to be able to exploit the performance of such systems. This book covers the scope of parallel programming for modern high performance computing systems. It first discusses selected and popular state-of-the-art computing devices and systems available today. These include multicore CPUs, manycore (co)processors, such as Intel Xeon Phi, accelerators, such as GPUs, and clusters, as well as programming models supported on these platforms. It next introduces parallelization through important programming paradigms, such as master-slave, geometric Single Program Multiple Data (SPMD) and divide-and-conquer. The practical and useful elements of the most popular and important APIs for programming parallel HPC systems are discussed, including MPI, OpenMP, Pthreads, CUDA, OpenCL, and OpenACC. It also demonstrates, through selected code listings, how selected APIs can be used to implement important programming paradigms. Furthermore, it shows how the codes can be compiled and executed in a Linux environment. The book also presents hybrid codes that integrate selected APIs for potentially multi-level parallelization and utilization of heterogeneous resources, and it shows how to use modern elements of these APIs. Selected optimization techniques are also included, such as overlapping communication and computations implemented using various APIs. Features: Discusses the popular and currently available computing devices and cluster systems Includes typical paradigms used in parallel programs Explores popular APIs for programming parallel applications Provides code templates that can be used for implementation of paradigms Provides hybrid code examples allowing multi-level parallelization Covers the optimization of parallel programs.
Summary OpenCL in Action is a thorough, hands-on presentation of OpenCL, with an eye toward showing developers how to build high-performance applications of their own. It begins by presenting the core concepts behind OpenCL, including vector computing, parallel programming, and multi-threaded operations, and then guides you step-by-step from simple data structures to complex functions. About the Technology Whatever system you have, it probably has more raw processing power than you're using. OpenCL is a high-performance programming language that maximizes computational power by executing on CPUs, graphics processors, and other number-crunching devices. It's perfect for speed-sensitive tasks like vector computing, matrix operations, and graphics acceleration. About this Book OpenCL in Action blends the theory of parallel computing with the practical reality of building high-performance applications using OpenCL. It first guides you through the fundamental data structures in an intuitive manner. Then, it explains techniques for high-speed sorting, image processing, matrix operations, and fast Fourier transform. The book concludes with a deep look at the all-important subject of graphics acceleration. Numerous challenging examples give you different ways to experiment with working code. A background in C or C++ is helpful, but no prior exposure to OpenCL is needed. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. What's Inside Learn OpenCL step by step Tons of annotated code Tested algorithms for maximum performance *********** Table of Contents PART 1 FOUNDATIONS OF OPENCL PROGRAMMING Introducing OpenCL Host programming: fundamental data structures Host programming: data transfer and partitioning Kernel programming: data types and device memory Kernel programming: operators and functions Image processing Events, profiling, and synchronization Development with C++ Development with Java and Python General coding principles PART 2 CODING PRACTICAL ALGORITHMS IN OPENCL Reduction and sorting Matrices and QR decomposition Signal processing and the fast Fourier transform PART 3 ACCELERATING OPENGL WITH OPENCL Combining OpenCL and OpenGL Textures and renderbuffers

This book mainly focuses on cloud security and high performance computing for cloud auditing. The book discusses emerging challenges and techniques developed for high performance semantic cloud auditing, and presents the state of the art in cloud auditing, computing and security techniques with focus on technical aspects and feasibility of auditing issues in federated cloud computing environments. In summer 2011, the United States Air Force Research Laboratory (AFRL) CyberBAT Cloud Security and Auditing Team initiated the exploration of the cloud security challenges and future cloud auditing research directions that are covered in this book. This work was supported by the United States government funds from the Air Force Office of Scientific Research (AFOSR), the AFOSR Summer Faculty Fellowship Program (SFFP), the Air Force Research Laboratory (AFRL) Visiting Faculty Research Program (VFRP), the National Science Foundation (NSF) and the National Institute of Health (NIH). All chapters were partially supported by the AFOSR Information Operations and Security Program extramural and intramural funds (AFOSR/RSL Program Manager: Dr. Robert Herklotz). Key Features: · Contains surveys of cyber threats and security issues in cloud computing and presents secure cloud architectures · Presents in-depth cloud auditing techniques, federated cloud security architectures, cloud access control models, and access assured information sharing technologies · Outlines a wide range of challenges and provides solutions to manage and control very large and complex data sets
Heterogeneous Computing with OpenCL, Second Edition teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs) such as AMD Fusion technology. It is the first textbook that presents OpenCL programming appropriate for the classroom and is intended to support a parallel programming course. Students will come away from this text with hands-on experience and significant knowledge of the syntax and use of OpenCL to address a range of fundamental parallel algorithms. Designed to work on multiple platforms and with wide industry support, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, Heterogeneous Computing with OpenCL explores memory spaces, optimization techniques, graphics interoperability, extensions, and debugging and profiling. It includes detailed examples throughout, plus additional online exercises and other supporting materials that can be downloaded at http://www.heterogeneouscompute.org/?page_id=7 This book will appeal to software engineers, programmers, hardware engineers, and students/advanced students. Explains principles and strategies to learn parallel programming with OpenCL, from understanding the four abstraction models to thoroughly testing and debugging complete applications. Covers image processing, web plugins, particle simulations, video editing, performance optimization, and more. Shows how OpenCL maps to an example target architecture and explains some of the tradeoffs associated with mapping to various architectures Addresses a range of fundamental programming techniques, with multiple examples and case studies that demonstrate OpenCL extensions for a variety of hardware platforms

OpenVX is the computer vision API adopted by many high-performance processor vendors. It is quickly becoming the
preferred way to write fast and power-efficient code on embedded systems. OpenVX Programming Guidebook presents
definitive information on OpenVX 1.2 and 1.3, the Neural Network, and other extensions as well as the OpenVX
Safety Critical standard. This book gives a high-level overview of the OpenVX standard, its design principles,
and overall structure. It covers computer vision functions and the graph API, providing examples of usage for the
majority of the functions. It is intended both for the first-time user of OpenVX and as a reference for
experienced OpenVX developers. Get to grips with the OpenVX standard and gain insight why various options were
chosen Start developing efficient OpenVX code instantly Understand design principles and use them to create
robust code Develop consumer and industrial products that use computer vision to understand and interact with the
real world

Embedded Systems: ARM Programming and Optimization combines an exploration of the ARM architecture with an
examination of the facilities offered by the Linux operating system to explain how various features of program
design can influence processor performance. It demonstrates methods by which a programmer can optimize program
code in a way that does not impact its behavior but improves its performance. Several applications, including
image transformations, fractal generation, image convolution, and computer vision tasks, are used to describe and
demonstrate these methods. From this, the reader will gain insight into computer architecture and application
design, as well as gain practical knowledge in the area of embedded software design for modern embedded systems.

This book constitutes the refereed post-conference proceedings of 13 workshops held at the 33rd International ISC
High Performance 2018 Conference, in Frankfurt, Germany, in June 2018: HPC I/O in the Data Center, HPC-IODC 2018;
Workshop on Performance and Scalability of Storage Systems, WOPSSS 2018; 13th Workshop on Virtualization in High-
Performance Cloud Computing, VHPC 2018; Third International Workshop on In Situ Visualization, WOIV 2018; 4th
International Workshop on Communication Architectures for HPC, Big Data, Deep Learning and Clouds at Extreme
Scale, ExaComm 2018; International Workshop on OpenPOWER for HPC, IWOPH 2018; IXPUG Workshop: Many-Core Computing
on Intel Processors; Workshop on Sustainable Ultrascale Computing Systems; Approximate and Transprecision
Computing on Emerging Technologies, ATCET 2018; First Workshop on the Convergence of Large-Scale Simulation and
Artificial Intelligence; Third Workshop for Open Source Supercomputing, OpenSuCo 2018; First Workshop on
Interactive High-Performance Computing; Workshop on Performance Portable Programming Models for Accelerators,
P^*3MA 2018. The 53 full papers included in this volume were carefully reviewed and selected from 80 submissions.
They cover all aspects of research, development, and application of large-scale, high performance experimental
and commercial systems. Topics include HPC computer architecture and hardware; programming models, system
software, and applications; solutions for heterogeneity, reliability, power efficiency of systems; virtualization and containerized environments; big data and cloud computing; and artificial intelligence.

This book follows an example-driven, simplified, and practical approach to using OpenCL for general purpose GPU programming. If you are a beginner in parallel programming and would like to quickly accelerate your algorithms using OpenCL, this book is perfect for you! You will find the diverse topics and case studies in this book interesting and informative. You will only require a good knowledge of C programming for this book, and an understanding of parallel implementations will be useful, but not necessary.

This book contains extended versions of the best papers presented at the 15th International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications, ICTERI 2019, held in Kherson, Ukraine, in June 2019. The 19 revised full papers included in this volume were carefully reviewed and selected from 416 initial submissions. The papers are organized in the following topical sections: advances in ICT and IS research; ICT in teaching, learning, and education management; applications of ICT in industrial and public practice.

Heterogeneous Computing with OpenCL 2.0 teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs). This fully-revised edition includes the latest enhancements in OpenCL 2.0 including: • Shared virtual memory to increase programming flexibility and reduce data transfers that consume resources • Dynamic parallelism which reduces processor load and avoids bottlenecks • Improved imaging support and integration with OpenGL Designed to work on multiple platforms, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, this book explores memory spaces, optimization techniques, extensions, debugging and profiling. Multiple case studies and examples illustrate high-performance algorithms, distributing work across heterogeneous systems, embedded domain-specific languages, and will give you hands-on OpenCL experience to address a range of fundamental parallel algorithms. Updated content to cover the latest developments in OpenCL 2.0, including improvements in memory handling, parallelism, and imaging support Explanations of principles and strategies to learn parallel programming with OpenCL, from understanding the abstraction models to thoroughly testing and debugging complete applications Example code covering image analytics, web plugins, particle simulations, video editing, performance optimization, and more

This book features selected papers presented at the Fifth International Conference on Nanoelectronics, Circuits and Communication Systems (NCCS 2019). It covers a range of topics, including nanoelectronic devices, microelectronics devices, material science, machine learning, Internet of things, cloud computing, computing systems, wireless communication systems, advances in communication 5G and beyond. Further, it discusses VLSI circuits and systems, MEMS, IC design and testing, electronic system design and manufacturing, speech signal processing, digital signal processing, FPGA-based wireless communication systems and FPGA-based system design.
Industry 4.0, e-farming, semiconductor memories, and IC fault detection and correction.

Computing Handbook, Third Edition: Computer Science and Software Engineering mirrors the modern taxonomy of computer science and software engineering as described by the Association for Computing Machinery (ACM) and the IEEE Computer Society (IEEE-CS). Written by established leading experts and influential young researchers, the first volume of this popular handbook examines the elements involved in designing and implementing software, new areas in which computers are being used, and ways to solve computing problems. The book also explores our current understanding of software engineering and its effect on the practice of software development and the education of software professionals. Like the second volume, this first volume describes what occurs in research laboratories, educational institutions, and public and private organizations to advance the effective development and use of computers and computing in today’s world. Research-level survey articles provide deep insights into the computing discipline, enabling readers to understand the principles and practices that drive computing education, research, and development in the twenty-first century.

Explore GPU-enabled programmable environment for machine learning, scientific applications, and gaming using PuCUDA, PyOpenGL, and Anaconda Accelerate Key Features Understand effective synchronization strategies for faster processing using GPUs Write parallel processing scripts with PyCuda and PyOpenCL Learn to use the CUDA libraries like CuDNN for deep learning on GPUs Book Description GPUs are proving to be excellent general purpose-parallel computing solutions for high performance tasks such as deep learning and scientific computing. This book will be your guide to getting started with GPU computing. It will start with introducing GPU computing and explain the architecture and programming models for GPUs. You will learn, by example, how to perform GPU programming with Python, and you’ll look at using integrations such as PyCUDA, PyOpenCL, CuPy and Numba with Anaconda for various tasks such as machine learning and data mining. Going further, you will get to grips with GPU work flows, management, and deployment using modern containerization solutions. Toward the end of the book, you will get familiar with the principles of distributed computing for training machine learning models and enhancing efficiency and performance. By the end of this book, you will be able to set up a GPU ecosystem for running complex applications and data models that demand great processing capabilities, and be able to efficiently manage memory to compute your application effectively and quickly. What you will learn Utilize Python libraries and frameworks for GPU acceleration Set up a GPU-enabled programmable machine learning environment on your system with Anaconda Deploy your machine learning system on cloud containers with illustrated examples Explore PyCUDA and PyOpenCL and compare them with platforms such as CUDA, OpenCL and ROCm. Perform data mining tasks with machine learning models on GPUs Extend your knowledge of GPU computing in scientific applications Who this book is for Data Scientist, Machine Learning enthusiasts and professionals who wants to get started with GPU computation and perform the complex tasks with low-latency. Intermediate knowledge of Python programming is assumed.

High Performance Parallelism Pearls shows how to leverage parallelism on processors and coprocessors with the
same programming – illustrating the most effective ways to better tap the computational potential of systems with Intel Xeon Phi coprocessors and Intel Xeon processors or other multicore processors. The book includes examples of successful programming efforts, drawn from across industries and domains such as chemistry, engineering, and environmental science. Each chapter in this edited work includes detailed explanations of the programming techniques used, while showing high performance results on both Intel Xeon Phi coprocessors and multicore processors. Learn from dozens of new examples and case studies illustrating "success stories" demonstrating not just the features of these powerful systems, but also how to leverage parallelism across these heterogeneous systems. Promotes consistent standards-based programming, showing in detail how to code for high performance on multicore processors and Intel® Xeon Phi™ Examples from multiple vertical domains illustrating parallel optimizations to modernize real-world codes Source code available for download to facilitate further exploration

This book brings together research on numerical methods adapted for Graphics Processing Units (GPUs). It explains recent efforts to adapt classic numerical methods, including solution of linear equations and FFT, for massively parallel GPU architectures. This volume consolidates recent research and adaptations, covering widely used methods that are at the core of many scientific and engineering computations. Each chapter is written by authors working on a specific group of methods; these leading experts provide mathematical background, parallel algorithms and implementation details leading to reusable, adaptable and scalable code fragments. This book also serves as a GPU implementation manual for many numerical algorithms, sharing tips on GPUs that can increase application efficiency. The valuable insights into parallelization strategies for GPUs are supplemented by ready-to-use code fragments. Numerical Computations with GPUs targets professionals and researchers working in high performance computing and GPU programming. Advanced-level students focused on computer science and mathematics will also find this book useful as secondary text book or reference.

The Cocoa programming environment—Apple’s powerful set of clean, object-oriented APIs—is increasingly becoming the basis of almost all contemporary Mac OS X development. With its long history of constant refinement and improvement, Cocoa has matured into a sophisticated programming environment that can make Mac OS X application development quick, efficient, and even fun. Yet for all its refined elegance and ease of use, the staggering size of the Cocoa family of APIs and the vast magnitude of the official documentation can be intimidating to even seasoned programmers. To help Mac OS X developers sort through and begin to put to practical use Cocoa’s vast array of tools and technologies, Cocoa Programming Developer’s Handbook provides a guided tour of the Cocoa APIs found on Mac OS X, thoroughly discussing—and showing in action—Cocoa's core frameworks and other vital components, as well as calling attention to some of the more interesting but often overlooked parts of the APIs and tools. This book provides expert insight into a wide range of key topics, from user interface design to network programming and performance tuning.

Intel® Xeon Phi™ Coprocessor Architecture and Tools: The Guide for Application Developers provides developers a comprehensive introduction and in-depth look at the Intel Xeon Phi coprocessor architecture and the corresponding
parallel data structure tools and algorithms used in the various technical computing applications for which it is suitable. It also examines the source code-level optimizations that can be performed to exploit the powerful features of the processor. Xeon Phi is at the heart of world’s fastest commercial supercomputer, which thanks to the massively parallel computing capabilities of Intel Xeon Phi processors coupled with Xeon Phi coprocessors attained 33.86 teraflops of benchmark performance in 2013. Extracting such stellar performance in real-world applications requires a sophisticated understanding of the complex interaction among hardware components, Xeon Phi cores, and the applications running on them. In this book, Rezaur Rahman, an Intel leader in the development of the Xeon Phi coprocessor and the optimization of its applications, presents and details all the features of Xeon Phi core design that are relevant to the practice of application developers, such as its vector units, hardware multithreading, cache hierarchy, and host-to-coprocessor communication channels. Building on this foundation, he shows developers how to solve real-world technical computing problems by selecting, deploying, and optimizing the available algorithms and data structure alternatives matching Xeon Phi’s hardware characteristics. From Rahman’s practical descriptions and extensive code examples, the reader will gain a working knowledge of the Xeon Phi vector instruction set and the Xeon Phi microarchitecture whereby cores execute 512-bit instruction streams in parallel.

In 2011 many computer users were exploring the opportunities and the benefits of the massive parallelism offered by heterogeneous computing. In 2000 the Khronos Group, a not-for-profit industry consortium, was founded to create standard open APIs for parallel computing, graphics and dynamic media. Among them has been OpenCL, an open system for programming heterogeneous computers with components made by multiple manufacturers. This publication explains how heterogeneous computers work and how to program them using OpenCL. It also describes how to combine OpenCL with OpenGL for displaying graphical effects in real time. Chapter 1 describes briefly two older de facto standard and highly successful parallel programming systems: MPI and OpenMP. Collectively, the MPI, OpenMP, and OpenCL systems cover programming of all major parallel architectures: clusters, shared-memory computers, and the newest heterogeneous computers. Chapter 2, the technical core of the book, deals with OpenCL fundamentals: programming, hardware, and the interaction between them. Chapter 3 adds important information about such advanced issues as double-versus-single arithmetic precision, efficiency, memory use, and debugging. Chapters 2 and 3 contain several examples of code and one case study on genetic algorithms. These examples are related to linear algebra operations, which are very common in scientific, industrial, and business applications. Most of the book’s examples can be found on the enclosed CD, which also contains basic projects for Visual Studio, MinGW, and GCC. This supplementary material will assist the reader in getting a quick start on OpenCL projects.

This book constitutes the refereed proceedings of the 16th International Conference on Integrated Formal Methods, IFM 2019, held in Lugano, Switzerland, in November 2020. The 24 full papers and 2 short papers were carefully reviewed and selected from 63 submissions. The papers cover a broad spectrum of topics: Integrating Machine Learning and Formal Modelling; Modelling and Verification in B and Event-B; Program Analysis and Testing; Verification of Interactive Behaviour; Formal Verification; Static Analysis; Domain-Specific Approaches; and
Algebraic Techniques.

Heterogeneous Computing with OpenCL, Second Edition teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs) such as AMD Fusion technology. It is the first textbook that presents OpenCL programming appropriate for the classroom and is intended to support a parallel programming course. Students will come away from this text with hands-on experience and significant knowledge of the syntax and use of OpenCL to address a range of fundamental parallel algorithms. Designed to work on multiple platforms and with wide industry support, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, Heterogeneous Computing with OpenCL explores memory spaces, optimization techniques, graphics interoperability, extensions, and debugging and profiling. It includes detailed examples throughout, plus additional online exercises and other supporting materials that can be downloaded at http://www.heterogeneouscompute.org/?page_id=7 This book will appeal to software engineers, programmers, hardware engineers, and students/advanced students. Explains principles and strategies to learn parallel programming with OpenCL, from understanding the four abstraction models to thoroughly testing and debugging complete applications. Covers image processing, web plugins, particle simulations, video editing, performance optimization, and more. Shows how OpenCL maps to an example target architecture and explains some of the tradeoffs associated with mapping to various architectures Addresses a range of fundamental programming techniques, with multiple examples and case studies that demonstrate OpenCL extensions for a variety of hardware platforms.

Using the new OpenCL (Open Computing Language) standard, you can write applications that access all available programming resources: CPUs, GPUs, and other processors such as DSPs and the Cell/B.E. processor. Already implemented by Apple, AMD, Intel, IBM, NVIDIA, and other leaders, OpenCL has outstanding potential for PCs, servers, handheld/embedded devices, high performance computing, and even cloud systems. This is the first comprehensive, authoritative, and practical guide to OpenCL 1.1 specifically for working developers and software architects. Written by five leading OpenCL authorities, OpenCL Programming Guide covers the entire specification. It reviews key use cases, shows how OpenCL can express a wide range of parallel algorithms, and offers complete reference material on both the API and OpenCL C programming language. Through complete case studies and downloadable code examples, the authors show how to write complex parallel programs that decompose workloads across many different devices. They also present all the essentials of OpenCL software performance optimization, including probing and adapting to hardware. Coverage includes Understanding OpenCL's architecture, concepts, terminology, goals, and rationale Programming with OpenCL C and the runtime API Using buffers, sub-buffers, images, samplers, and events Sharing and synchronizing data with OpenGL and Microsoft’s Direct3D Simplifying development with the C++ Wrapper API Using OpenCL Embedded Profiles to support devices ranging from cellphones to supercomputer nodes Case studies dealing with physics simulation; image and signal processing, such as image
histograms, edge detection filters, Fast Fourier Transforms, and optical flow; math libraries, such as matrix multiplication and high-performance sparse matrix multiplication; and more Source code for this book is available at [https://code.google.com/p/opencl-book-samples/](https://code.google.com/p/opencl-book-samples/)

This book constitutes the thoroughly refereed post-conference proceedings of the 7th International Haifa Verification Conference, HVC 2011, held in Haifa, Israel in December 2011. The 15 revised full papers presented together with 3 tool papers and 4 posters were carefully reviewed and selected from 43 submissions. The papers are organized in topical sections on synthesis, formal verification, software quality, testing and coverage, experience and tools, and posters- student event.

This accessible textbook/reference reviews the fundamental concepts and practical issues involved in designing digital surveillance systems that fully exploit the power of intelligent computing techniques. The book presents comprehensive coverage of all aspects of such systems, from camera calibration and data capture, to the secure transmission of surveillance data. In addition to the detection and recognition of objects and biometric features, the text also examines the automated observation of surveillance events, and how this can be enhanced through the use of deep learning methods and supercomputing technology. This updated new edition features extended coverage on face detection, pedestrian detection and privacy preservation for intelligent surveillance. Topics and features: contains review questions and exercises in every chapter, together with a glossary; describes the essentials of implementing an intelligent surveillance system and analyzing surveillance data, including a range of biometric characteristics; examines the importance of network security and digital forensics in the communication of surveillance data, as well as issues of privacy and ethics; discusses the Viola-Jones object detection method, and the HOG algorithm for pedestrian and human behavior recognition; reviews the use of artificial intelligence for automated monitoring of surveillance events, and decision-making approaches to determine the need for human intervention; presents a case study on a system that triggers an alarm when a vehicle fails to stop at a red light, and identifies the vehicle’s license plate number; investigates the use of cutting-edge supercomputing technologies for digital surveillance, such as FPGA, GPU and parallel computing. This concise, classroom-tested textbook is ideal for undergraduate and postgraduate-level courses on intelligent surveillance. Researchers interested in entering this area will also find the book suitable as a helpful self-study reference.

Programming is now parallel programming. Much as structured programming revolutionized traditional serial programming decades ago, a new kind of structured programming, based on patterns, is relevant to parallel programming today. Parallel computing experts and industry insiders Michael McCool, Arch Robison, and James Reinders describe how to design and implement maintainable and efficient parallel algorithms using a pattern-based approach. They present both theory and practice, and give detailed concrete examples using multiple programming models. Examples are primarily given using two of the most popular and cutting edge programming
models for parallel programming: Threading Building Blocks, and Cilk Plus. These architecture-independent models enable easy integration into existing applications, preserve investments in existing code, and speed the development of parallel applications. Examples from realistic contexts illustrate patterns and themes in parallel algorithm design that are widely applicable regardless of implementation technology. The patterns-based approach offers structure and insight that developers can apply to a variety of parallel programming models. Develops a composable, structured, scalable, and machine-independent approach to parallel computing. Includes detailed examples in both Cilk Plus and the latest Threading Building Blocks, which support a wide variety of computers.

Heterogeneous Computing Architectures: Challenges and Vision provides an updated vision of the state-of-the-art of heterogeneous computing systems, covering all the aspects related to their design: from the architecture and programming models to hardware/software integration and orchestration to real-time and security requirements. The transitions from multicore processors, GPU computing, and Cloud computing are not separate trends, but aspects of a single trend-mainstream; computers from desktop to smartphones are being permanently transformed into heterogeneous supercomputer clusters. The reader will get an organic perspective of modern heterogeneous systems and their future evolution.

The four-volume set LNCS 11334-11337 constitutes the proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing, ICA3PP 2018, held in Guangzhou, China, in November 2018. The 141 full and 50 short papers presented were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on Distributed and Parallel Computing; High Performance Computing; Big Data and Information Processing; Internet of Things and Cloud Computing; and Security and Privacy in Computing.

GPU Parallel Program Development using CUDA teaches GPU programming by showing the differences among different families of GPUs. This approach prepares the reader for the next generation and future generations of GPUs. The book emphasizes concepts that will remain relevant for a long time, rather than concepts that are platform-specific. At the same time, the book also provides platform-dependent explanations that are as valuable as generalized GPU concepts. The book consists of three separate parts; it starts by explaining parallelism using CPU multi-threading in Part I. A few simple programs are used to demonstrate the concept of dividing a large task into multiple parallel sub-tasks and mapping them to CPU threads. Multiple ways of parallelizing the same task are analyzed and their pros/cons are studied in terms of both core and memory operation. Part II of the book introduces GPU massive parallelism. The same programs are parallelized on multiple Nvidia GPU platforms and the same performance analysis is repeated. Because the core and memory structures of CPUs and GPUs are different, the results differ in interesting ways. The end goal is to make programmers aware of all the good ideas, as well as the bad ideas, so readers can apply the good ideas and avoid the bad ideas in their own programs. Part III of the book provides pointer for readers who want to expand their horizons. It provides a brief introduction to popular CUDA libraries (such as cuBLAS, cuFFT, NPP, and Thrust), the OpenCL programming language, an overview of GPU programming using other programming languages and API libraries (such as Python, OpenCV, OpenGL, and Apple’s
Swift and Metal,) and the deep learning library cuDNN.

The six-volume set LNCS 8579-8584 constitutes the refereed proceedings of the 14th International Conference on Computational Science and Its Applications, ICCSA 2014, held in Guimarães, Portugal, in June/July 2014. The 347 revised papers presented in 30 workshops and a special track were carefully reviewed and selected from 1167. The 289 papers presented in the workshops cover various areas in computational science ranging from computational science technologies to specific areas of computational science such as computational geometry and security.

The set of papers collected in this issue originated from the AGERE! Workshop series - the last edition was held in 2017 - and concern the application of actor-based approaches to mainstream application domains and the discussion of related issues. The issue is divided into two parts. The first part concerns Web Programming; Data-Intensive Parallel Programming; Mobile Computing; Self-Organizing Systems and the second part concerns Scheduling; Debugging; Communication and Coordination; Monitoring.


Copyright code: e52e643759c82f71c8ac2ccf8ffbe837