Cache Memory Book The Second Edition The Morgan Kaufmann Series In Computer Architecture And Design

How can you help your Drupal website continue to perform at the highest level as it grows to meet demand? This comprehensive guide provides best practices, examples, and in-depth explanations for solving several performance and scalability issues. You'll learn how to apply coding and infrastructure techniques to Drupal internals, application performance, databases, web servers, and performance analysis. Covering Drupal versions 7 and 8, this book is the ideal reference for everything from site deployment to implementing specific technologies such as Varnish, memcache, or Solr. If you have a basic understanding of Drupal and the Linux-Apache-MySQL-PHP (LAMP) stack, you're ready to get started. Establish a performance baseline and define goals for improvement Optimize your website's code and front-end performance Get best and worst practices for customizing Drupal core functionality Apply infrastructure design techniques to launch or expand a site Use tools to configure, monitor, and optimize MySQL performance Employ alternative storage and backend search options as your site grows Tune your web servers through httpd and PHP configuration Monitor services and perform load tests to catch problems before they become critical

After authoring a best-selling text in India, Dhananjay Dhamdhere has written Operating Systems, and it includes precise definitions and clear explanations of fundamental concepts, which makes this text an excellent text for the first course in operating systems.Concepts, techniques, and case studies are well integrated so many design and implementation details look obvious to the student. Exceptionally clear explanations of concepts are offered, and coverage of both fundamentals and such cutting-edge material like encryption and security is included. The numerous case studies are tied firmly to real-world experiences with operating systems that students will likely encounter.

The era of seemingly unlimited growth in processor performance is over: single chip architectures can no longer overcome the performance limitations imposed by the power they consume and the heat they generate. Today, Intel and other semiconductor firms are abandoning the single fast processor model in favor of multi-core microprocessors--chips that combine two or more processors in a single package. In the fourth edition of Computer Architecture, the authors focus on this historic shift, increasing their coverage of multiprocessors and exploring the most effective ways of achieving parallelism as the key to unlocking the power of multiple processor architectures. Additionally, the new edition has expanded and updated coverage of design topics beyond processor performance, including power, reliability, availability, and dependability. CD System Requirements PDF Viewer The CD material includes PDF documents that you can read with a PDF viewer such as Adobe, Acrobat or Adobe Reader. Recent versions of Adobe Reader for some platforms are included on the CD. HTML Browser The navigation framework on this CD is delivered in HTML and JavaScript. It is recommended that you install the latest version of your favorite HTML browser to view this CD. The content has been verified under Windows XP with the following browsers: Internet Explorer 6.0, Firefox 1.5; under Mac OS X (Panther) with the following browsers: Internet Explorer 5.2, Firefox 1.0.6, Safari 1.3; and under Mandriva Linux 2006 with the following browsers: Firefox 1.0.6, Konqueror 3.4.2, Mozilla 1.7.11. The content is designed to be viewed in a browser window that is at least 720 pixels wide. You may find the content does not display well if your display is not set to at least 1024x768 pixel resolution. Operating System This CD can be used under any operating system that includes an HTML browser and a PDF viewer. This includes Windows, Mac OS, and most Linux and Unix systems. Increased coverage on achieving parallelism with multiprocessors. Case studies of latest technology from industry including the Sun Niagara Multiprocessor, AMD Opteron, and Pentium 4. Three review appendices, included in the printed volume, review the basic and intermediate principles the main text relies upon. Eight reference appendices, collected on the CD, cover a range of topics including specific architectures, embedded systems, application specific processors--some guest authored by subject experts.

The goal of this book is to present and compare various options one for systems architecture from two separate points of view. One, that of the information technology decision-maker who must choose a solution matching company business requirements, and secondly that of the systems architect who finds himself between the rock of changes in hardware and software technologies and the hard place of changing business needs. Different aspects of server architecture are presented, from databases designed for parallel architectures to high-availability systems, and touching en route on often- neglected performance aspects. The book provides IT managers, decision makers and project leaders who want to acquire knowledge sufficient to understand the choices made in and capabilities of systems offered by various vendors Provides system design information to balance the characteristic applications against the capabilities and nature of various architectural choices In addition, it offers an integrated view of the concepts in server architecture, accompanied by discussion of effects on the evolution of the data processing industry

The Second Edition of The Cache Memory Book introduces systems designers to the concepts behind cache design. The book teaches the basic cache concepts and more exotic techniques. It leads readers through someof the most intricate protocols used in complex multiprocessor caches. Written in an accessible, informal style, this text demystifies cache memory design by translating cache concepts and jargon into practical methodologies and real-life examples. It also provides adequate detail to serve as a reference book for ongoing work in cache memory design. The Second Edition includes an updated and expanded glossary of cache memory terms and buzzwords. The book provides new real world applications of cache memory design and a new chapter on cache"tricks". Illustrates detailed example designs of caches Provides numerous examples in the form of block diagrams, timing waveforms, state tables, and code traces Defines and discusses more than 240 cache specific buzzwords, comparing in detail the relative merits of different design methodologies Includes an extensive glossary, complete with clear definitions, synonyms, and references to the appropriate text discussions

Many modern computer systems and most multicore chips (chip multiprocessors) support shared memory in hardware. In a shared memory system, each of the processor cores may read and write to a single shared address space. For a shared memory machine, the memory consistency model defines the architecturally visible behavior of its memory system. Consistency definitions provide rules about loads and stores (or memory reads and writes) and how they act upon memory. As part of supporting a memory consistency model, many machines also provide cache coherence protocols that ensure that multiple cached copies of data are kept up-to-date. The goal of this primer is to provide readers with a basic understanding of consistency and coherence. This understanding includes both the issues that must be solved as well as a variety of solutions. We present both highlevel concepts as well as specific, concrete examples from real-world systems. Table of Contents: Preface / Introduction to Consistency and Coherence / Coherence Basics / Memory Consistency Motivation and Sequential Consistency / Total Store Order and the x86 Memory Model / Relaxed Memory Consistency / Coherence Protocols / Snooping Coherence Protocols / Directory Coherence Protocols / Advanced Topics in Coherence / Author Biographies

The Cache Memory BookMorgan Kaufmann

A key determinant of overall system performance and power dissipation is the cache hierarchy since access to off-chip memory consumes many more cycles and energy than on-chip accesses. In addition, multi-core processors are expected to place ever higher bandwidth demands on the memory system. All these issues make it important to avoid off-chip memory access by improving the efficiency of the on-chip cache. Future multi-core processors will have many large cache banks connected by a network and shared by many cores. Hence, many important problems must be solved: cache resources must be allocated across many cores, data must be placed in cache banks that are near the accessing core, and the most important data must be identified for retention. Finally, difficulties in scaling existing technologies

require adapting to and exploiting new technology constraints. The book attempts a synthesis of recent cache research that has focused on innovations for multi-core processors. It is an excellent starting point for early-stage graduate students, researchers, and practitioners who wish to understand the landscape of recent cache research. The book is suitable as a reference for advanced computer architecture classes as well as for experienced researchers and VLSI engineers. Table of Contents: Basic Elements of Large Cache Design / Organizing Data in CMP Last Level Caches / Policies Impacting Cache Hit Rates / Interconnection Networks within Large Caches / Technology / Concluding Remarks

Beginning and experienced programmers will use this comprehensive guide to persistent memory programming. You will understand how persistent memory brings together several new software/hardware requirements, and offers great promise for better performance and faster application startup times—a huge leap forward in byte-addressable capacity compared with current DRAM offerings. This revolutionary new technology gives applications significant performance and capacity improvements over existing technologies. It requires a new way of thinking and developing, which makes this highly disruptive to the IT/computing industry. The full spectrum of industry sectors that will benefit from this technology include, but are not limited to, in-memory and traditional databases, AI, analytics, HPC, virtualization, and big data. Programming Persistent Memory describes the technology and why it is exciting the industry. It covers the operating system and hardware requirements as well as how to create development environments using emulated or real persistent memory hardware. The book explains fundamental concepts; provides an introduction to persistent memory programming APIs for C, C++, JavaScript, and other languages; discusses RMDA with persistent memory; reviews security features; and presents many examples. Source code and examples that you can run on your own systems are included. What You'll Learn Understand what persistent memory is, what it does, and the value it brings to the industry Become familiar with the operating system and hardware requirements to use persistent memory Know the fundamentals of persistent memory programming: why it is different from current programming methods, and what developers need to keep in mind when programming for persistence Look at persistent memory application development by example using the Persistent Memory Development Kit (PMDK)Design and optimize data structures for persistent memoryStudy how real-world applications are modified to leverage persistent memoryUtilize the tools available for persistent memory programming, application performance profiling, and debugging Who This Book Is For C, C++, Java, and Python developers, but will also be useful to software, cloud, and hardware architects across a broad spectrum of sectors, including cloud service providers, independent software vendors, high performance compute, artificial intelligence, data analytics, big data, etc.

The first in-depth, complete, and unified theoretical discussion of the two most important classes of algorithms for solving matrix eigenvalue problems: QR-like algorithms for dense problems and Krylov subspace methods for sparse problems. The author discusses the theory of the generic GR algorithm, including special cases (for example, QR, SR, HR), and the development of Krylov subspace methods. This book also addresses a generic Krylov process and the Arnoldi and various Lanczos algorithms, which are obtained as special cases. Theoretical and computational exercises guide students, step by step, to the results. Downloadable MATLAB programs, compiled by the author, are available on a supplementary Web site. Readers of this book are expected to be familiar with the basic ideas of linear algebra and to have had some experience with matrix computations. Ideal for graduate students, or as a reference book for researchers and users of eigenvalue codes.

Over the last ten years, the ARM architecture has become one of the most pervasive architectures in the world, with more than 2 billion ARM-based processors embedded in products ranging from cell phones to automotive braking systems. A world-wide community of ARM developers in semiconductor and product design companies includes software developers, system designers and hardware engineers. To date no book has directly addressed their need to develop the system and software for an ARM-based system. This text fills that gap. This book provides a comprehensive description of the operation of the ARM core from a developer's perspective with a clear emphasis on software. It demonstrates not only how to write efficient ARM software in C and assembly but also how to optimize code. Example code throughout the book can be integrated into commercial products or used as templates to enable quick creation of productive software. The book covers both the ARM and Thumb instruction sets, covers Intel's XScale Processors, outlines distinctions among the versions of the ARM architecture, demonstrates how to implement DSP algorithms, explains exception and interrupt handling, describes the cache technologies that surround the ARM cores as well as the most efficient memory management techniques. A final chapter looks forward to the future of the ARM architecture considering ARMv6, the latest change to the instruction set, which has been designed to improve the DSP and media processing capabilities of the architecture. * No other book describes the ARM core from a system and software perspective. * Author team combines extensive ARM software engineering experience with an in-depth knowledge of ARM developer needs. * Practical, executable code is fully explained in the book and available on the publisher's Website. * Includes a simple embedded operating system.

Is your memory hierarchy stopping your microprocessor from performing at the high level it should be? Memory Systems: Cache, DRAM, Disk shows you how to resolve this problem. The book tells you everything you need to know about the logical design and operation, physical design and operation, performance characteristics and resulting design trade-offs, and the energy consumption of modern memory hierarchies. You learn how to to tackle the challenging optimization problems that result from the side-effects that can appear at any point in the entire hierarchy. As a result you will be able to design and emulate the entire memory hierarchy. Understand all levels of the system hierarchy -Xcache, DRAM, and disk. Evaluate the system-level effects of all design choices. Model performance and energy consumption for each component in the memory hierarchy.

Takes a unique systems approach to programming and architecture of the VAX Using the VAX as a detailed example, the first half of this book offers a complete course in assembly language programming. The second describes higher-level systems issues in computer architecture. Highlights include the VAX assembler and debugger, other modern architectures such as RISCs, multiprocessing and parallel computing, microprogramming, caches and translation buffers, and an appendix on the Berkeley UNIX assembler.

An Essential Reference for Intermediate and Advanced R Programmers Advanced R presents useful tools and techniques for attacking many types of R programming problems, helping you avoid mistakes and dead ends. With more than ten years of experience programming in R, the author illustrates the elegance, beauty, and flexibility at the heart of R. The book develops the necessary skills to produce quality code that can be used in a variety of circumstances. You will learn: The fundamentals of R, including standard data types and functions Functional programming as a useful framework for solving wide classes of problems The positives and negatives of metaprogramming How to write fast, memory-efficient code This book not only helps current R users become R programmers but also shows existing programmers what's special about R. Intermediate R programmers can

dive deeper into R and learn new strategies for solving diverse problems while programmers from other languages can learn the details of R and understand why R works the way it does.

How long does it take for your website to load? Web performance is just as critical for small and medium-sized websites as it is for massive websites that receive tons of hits. Before you pour money and time into rewriting your code or replacing your infrastructure, first consider a reverse-caching proxy server like Varnish. With this practical book, you'll learn how Varnish can give your website or API an immediate performance boost. Varnish mimicks the behavior of your webserver, caches its output in memory, and serves the result directly to clients without having to access your webserver. If you're a web developer familiar with HTTP, this book helps you master Varnish basics, so you can get up and running in no time. You'll learn how to use the Varnish Configuration Language and HTTP best practices to achieve faster performance and a higher hit rate. Understand how Varnish helps you gain optimum web performance Use HTTP to improve the cache-ability of your websites, web applications, and APIs Properly invalidate your cache when the origin data changes Optimize access to your backend servers Avoid common mistakes when using Varnish in the wild Use logging and debugging tools to examine the behavior of Varnish

This synthesis lecture presents the current state-of-the-art in applying low-latency, lossless hardware compression algorithms to cache, memory, and the memory/cache link. There are many non-trivial challenges that must be addressed to make data compression work well in this context. First, since compressed data must be decompressed before it can be accessed, decompression latency ends up on the critical memory access path. This imposes a significant constraint on the choice of compression algorithms. Second, while conventional memory systems store fixed-size entities like data types, cache blocks, and memory pages, these entities will suddenly vary in size in a memory system that employs compression. Dealing with variable size entities in a memory system using compression has a significant impact on the way caches are organized and how to manage the resources in main memory. We systematically discuss solutions in the open literature to these problems. Chapter 2 provides the foundations of data compression by first introducing the fundamental concept of value locality. We then introduce a taxonomy of compression algorithms and show how previously proposed algorithms fit within that logical framework. Chapter 3 discusses the different ways that cache memory systems can employ compression, focusing on the trade-offs between latency, capacity, and complexity of alternative ways to compact compressed cache blocks. Chapter 4 discusses issues in applying data compression to main memory and Chapter 5 covers techniques for compressing data on the cache-to-memory links. This book should help a skilled memory system designer understand the fundamental challenges in applying compression to the memory hierarchy and introduce him/her to the state-of-the-art techniques in addressing them.

Many modern computer systems, including homogeneous and heterogeneous architectures, support shared memory in hardware. In a shared memory system, each of the processor cores may read and write to a single shared address space. For a shared memory machine, the memory consistency model defines the architecturally visible behavior of its memory system. Consistency definitions provide rules about loads and stores (or memory reads and writes) and how they act upon memory. As part of supporting a memory consistency model, many machines also provide cache coherence protocols that ensure that multiple cached copies of data are kept up-to-date. The goal of this primer is to provide readers with a basic understanding of consistency and coherence. This understanding includes both the issues that must be solved as well as a variety of solutions. We present both high-level concepts as well as specific, concrete examples from real-world systems. This second edition reflects a decade of advancements since the first edition and includes, among other more modest changes, two new chapters: one on consistency and coherence for non-CPU accelerators (with a focus on GPUs) and one that points to formal work and tools on consistency and coherence.

This book describes the architecture of microprocessors from simple in-order short pipeline designs to out-of-order superscalars.

This book constitutes the refereed proceedings of the 6th International Conference on Applied Parallel Computing, PARA 2002, held in Espoo, Finland, in June 2002. The 50 revised full papers presented together with nine keynote lectures were carefully reviewed and selected for inclusion in the proceedings. The papers are organized in topical sections on data mining and knowledge discovery, parallel program development, practical experience in parallel computing, computer science, numerical algorithms with hierarchical memory optimization, numerical methods and algorithms, cluster computing, grid and network technologies, and physics and applications.

The book Operating System by Rohit Khurana is an insightful work that elaborates on fundamentals as well as advanced topics of the discipline. It offers an in-depth coverage of concepts, design and functions of an operating system irrespective of the hardware used. With illustrations and examples the aim is to make the subject crystal clear and the book extremely student-friendly. The book caters to undergraduate students of most Indian universities, who would find subject matter highly informative and enriching. Tailored as a guide for self-paced learning, it equips budding system programmers with the right knowledge and expertise. The book has been revised to keep pace with the latest technology and constantly revising syllabuses. Thus, this edition has become more comprehensive with the inclusion of several new topics. In addition, certain sections of the book have been thoroughly revised. Key Features • Case studies of Unix, Linux and Windows to put theory concepts into practice • A crisp summary for recapitulation with each chapter • A glossary of technical terms • Insightful questions and model test papers to prepare for the examinations New in this Edition • More types of operating system, like PC and mobile; Methods used for communication in client-server systems. • New topics like: Thread library; Thread scheduling; Principles of concurrency, Precedence graph, Concurrency conditions and Sleeping barber problem; Structure of page tables, Demand segmentation and Cache memory organization; STREAMS; Disk attachment, Stable and tertiary storage, Record blocking and File sharing; Goals and principles of protection, Access control matrix, Revocation of access rights, Cryptography, Trusted systems, and Firewalls.

Intelligent readers who want to build their own embedded computer systems-- installed in everything from cell phones to cars to handheld organizers to refrigerators-- will find this book to be the most in-depth, practical, and up-to-date guide on the market. Designing Embedded Hardware carefully steers between the practical and philosophical aspects, so developers can both create their own devices and gadgets and customize and extend off-the-shelf systems. There are hundreds of books to choose from if you need to learn programming, but only a few are available if you want to learn to create hardware. Designing Embedded Hardware provides software and hardware engineers with no prior experience in embedded systems with the necessary conceptual and design building blocks to understand the architectures of embedded systems. Written to provide the depth of coverage and real-world examples developers need, Designing Embedded Hardware also provides a road-map to the pitfalls and traps to avoid in designing embedded systems. Designing Embedded Hardware covers such essential topics as: The principles of

developing computer hardware Core hardware designs Assembly language concepts Parallel I/O Analog-digital conversion Timers (internal and external) UART Serial Peripheral Interface Inter-Integrated Circuit Bus Controller Area Network (CAN) Data Converter Interface (DCI) Low-power operation This invaluable and eminently useful book gives you the practical tools and skills to develop, build, and program your own application-specific computers.

PC Hardware in a Nutshell is the practical guide to buying, building, upgrading, and repairing Intel-based PCs. A longtime favorite among PC users, the third edition of the book now contains useful information for people running either Windows or Linux operating systems. Written for novices and seasoned professionals alike, the book is packed with useful and unbiased information, including how-to advice for specific components, ample reference material, and a comprehensive case study on building a PC.In addition to coverage of the fundamentals and general tips about working on PCs, the book includes chapters focusing on motherboards, processors, memory, floppies, hard drives, optical drives, tape devices, video devices, input devices, audio components, communications, power supplies, and maintenance. Special emphasis is given to upgrading and troubleshooting existing equipment so you can get the most from your existing investments.This new edition is expanded to include: Detailed information about the latest motherboards and chipsets from AMD, Intel, SiS, and VIA Extensive coverage of the Pentium 4 and the latest AMD processors, including the Athlon XP/MP Full details about new hard drive standards, including the latest SCSI standards, ATA/133, Serial ATA, and the new 48-bit "Big Drive" ATA interface Extended coverage of DVD drives, including DVD-RAM, DVD-R/RW, and DVD+R/RW Details about Flat Panel Displays, including how to choose one (and why you might not want to) New chapters on serial communications, parallel communications, and USB communications (including USB 2.0) Enhanced troubleshooting coverage PC Hardware in a Nutshell, 3rd Edition provides independent, useful and practical information in a no-nonsense manner with specific recommendations on components. Based on real-world testing over time, it will help you make intelligent, informed decisions about buying, building, upgrading, and repairing PCs in a cost effective manner that will help you maximize new or existing computer hardware systems. It's loaded with real-world advice presented in a concise style that clearly delivers just the information you want, without your having to hunt for it.

General-purpose graphics processing units (GPGPU) have emerged as an important class of shared memory parallel processing architectures, with widespread deployment in every computer class from high-end supercomputers to embedded mobile platforms. Relative to more traditional multicore systems of today, GPGPUs have distinctly higher degrees of hardware multithreading (hundreds of hardware thread contexts vs. tens), a return to wide vector units (several tens vs. 1-10), memory architectures that deliver higher peak memory bandwidth (hundreds of gigabytes per second vs. tens), and smaller caches/scratchpad memories (less than 1 megabyte vs. 1-10 megabytes).In this book, we provide a high-level overview of current GPGPU architectures and programming models. We review the principles that are used in previous shared memory parallel platforms, focusing on recent results in both the theory and practice of parallel algorithms, and suggest a connection to GPGPU platforms. We aim to provide hints to architects about understanding algorithm aspect to GPGPU. We also provide detailed performance analysis and guide optimizations from high-level algorithms to low-level instruction level optimizations. As a case study, we use n-body particle simulations known as the fast multipole method (FMM) as an example. We also briefly survey the state-of-the-art in GPU performance analysis tools and techniques.

This textbook is designed for the first course in Computer Architecture, usually offered at the junior/senior (3rd, 4th year) level in electrical engineering, computer science or computer engineering departments. This course is required of all electrical engineering and computer science/computer engineering majors specializing in the design of computer systems. This text provides a comprehensive introduction to computer architecture, covering topic from design of simple microprocessors to techniques used in the most advanced supercomputers.

This lecture presents a study of the microarchitecture of contemporary microprocessors. The focus is on implementation aspects, with discussions on their implications in terms of performance, power, and cost of state-of-the-art designs. The lecture starts with an overview of the different types of microprocessors and a review of the microarchitecture of cache memories. Then, it describes the implementation of the fetch unit, where special emphasis is made on the required support for branch prediction. The next section is devoted to instruction decode with special focus on the particular support to decoding x86 instructions. The next chapter presents the allocation stage and pays special attention to the implementation of register renaming. Afterward, the issue stage is studied. Here, the logic to implement out-of-order issue for both memory and non-memory instructions is thoroughly described. The following chapter focuses on the instruction execution and describes the different functional units that can be found in contemporary microprocessors, as well as the implementation of the bypass network, which has an important impact on the performance. Finally, the lecture concludes with the commit stage, where it describes how the architectural state is updated and recovered in case of exceptions or misspeculations. This lecture is intended for an advanced course on computer architecture, suitable for graduate students or senior undergrads who want to specialize in the area of computer architecture. It is also intended for practitioners in the industry in the area of microprocessor design. The book assumes that the reader is familiar with the main concepts regarding pipelining, out-of-order execution, cache memories, and virtual memory. Table of Contents: Introduction / Caches / The Instruction Fetch Unit / Decode / Allocation / The Issue Stage / Execute / The Commit Stage / References / Author Biographies

Concurrency can be notoriously difficult to get right, but fortunately, the Go open source programming language makes working with concurrency tractable and even easy. If you’re a developer familiar with Go, this practical book demonstrates best practices and patterns to help you incorporate concurrency into your systems. Author Katherine Cox-Buday takes you step-by-step through the process. You’ll understand how Go chooses to model concurrency, what issues arise from this model, and how you can compose primitives within this model to solve problems. Learn the skills and tooling you need to confidently write and implement concurrent systems of any size. Understand how Go addresses fundamental problems that make concurrency difficult to do correctly Learn the key differences between concurrency and parallelism Dig into the syntax of Go’s memory synchronization primitives Form patterns with these primitives to write maintainable concurrent code Compose patterns into a series of practices that enable you to write large, distributed

systems that scale Learn the sophistication behind goroutines and how Go’s runtime stitches everything together
Graduate Aptitude Test in Engineering (GATE) is one of the recognized national level examinations that demands focussed study along with forethought, systematic planning and exactitude. Postgraduate Engineering Common Entrance Test (PGECET) is also one of those examinations, a student has to face to get admission in various postgraduate programs. So, in order to become up to snuff for this eligibility clause (qualifying GATE/PGECET), a student facing a very high competition should excel his/her standards to success by way of preparing from the standard books. This book guides students via simple, elegant and explicit presentation that blends theory logically and rigorously with the practical aspects bearing on computer science and information technology. The book not only keeps abreast of all the chapterwise information generally asked in the examinations but also proffers felicitous tips in the furtherance of problem-solving technique. HIGHLIGHTS OF THE BOOK • Systematic discussion of concepts endowed with ample illustrations • Notes are incorporated at several places giving additional information on the key concepts • Inclusion of solved practice exercises for verbal and numerical aptitude to guide students from practice and examination point of view • Prodigious objective-type questions based on the past years’ GATE examination questions with answer keys and in-depth explanation are available at https://www.phindia.com/GATE_AND_PGECET • Every solution lasts with a reference, thus providing a scope for further study The book, which will prove to be an epitome of learning the concepts of CS and IT for GATE/PGECET examination, is purely intended for the aspirants of GATE and PGECET examinations. It should also be of considerable utility and worth to the aspirants of UGC-NET as well as to those who wish to pursue career in public sector units like ONGC, NTPC, ISRO, BHEL, BARC, DRDO, DVC, Power-grid, IOCL and many more. In addition, the book is also of immense use for the placement coordinators of GATE/PGECET. TARGET AUDIENCE • GATE/PGECET Examination • UGC-NET Examination • Examinations conducted by PSUs like ONGC, NTPC, ISRO, BHEL, BARC, DRDO, DVC, Power-grid, IOCL and many more
One of the main concerns for digital photographers today is asset management: how to file, find, protect, and re-use their photos. The best solutions can be found in The DAM Book, our bestselling guide to managing digital images efficiently and effectively. Anyone who shoots, scans, or stores digital photographs is practicing digital asset management (DAM), but few people do it in a way that makes sense. In this second edition, photographer Peter Krogh -- the leading expert on DAM -- provides new tools and techniques to help professionals, amateurs, and students: Understand the image file lifecycle: from shooting to editing, output, and permanent storage Learn new ways to use metadata and key words to track photo files Create a digital archive and name files clearly Determine a strategy for backing up and validating image data Learn a catalog workflow strategy, using Adobe Bridge, Camera Raw, Adobe Lightroom, Microsoft Expression Media, and Photoshop CS4 together Migrate images from one file format to another, from one storage medium to another, and from film to digital Learn how to copyright images To identify and protect your images in the marketplace, having a solid asset management system is essential. The DAM Book offers the best approach.
This title gives students an integrated and rigorous picture of applied computer science, as it comes to play in the construction of a simple yet powerful computer system.
A stunning array of full-color photographs captures the history of modern technology through images of the computer collection of the Computer History Museum in Silicon Valley, offering revealing glimpses of such seminal machines as the Eniac, Crays 1-3, and Apple I and II, while describing each model, their innovations, and place in computer history.
Exam Board: OCR Level: GCSE Subject: Computer Science First Teaching: September 2016 First Exam: June 2018 Build student confidence and ensure successful progress through GCSE Computer Science. Our expert authors provide insight and guidance to meet the demands of the new OCR specification, with challenging tasks and activities to test the computational skills and knowledge required for success in their exams, and advice for successful completion of the non-examined assessment. - Builds students' knowledge and confidence through detailed topic coverage and explanation of key terms - Develops computational thinking skills with practice exercises and problem-solving tasks - Ensures progression through GCSE with regular assessment questions, that can be developed with supporting Dynamic Learning digital resources - Instils a deeper understanding and awareness of computer science, and its applications and implications in the wider world
An authoritative book for hardware and software designers. Caches are by far the simplest and most effective mechanism for improving computer performance. This innovative book exposes the characteristics of performance-optimal single and multi-level cache hierarchies by approaching the cache design process through the novel perspective of minimizing execution times. It presents useful data on the relative performance of a wide spectrum of machines and offers empirical and analytical evaluations of the underlying phenomena. This book will help computer professionals appreciate the impact of caches and enable designers to maximize performance given particular implementation constraints.
Many bookstores offer numerous choices of books on Java Server Programming; however, most of these books are intricate and complex to grasp. So, what are your chances of picking up the right one? If this question has been troubling you, be rest assured now! This book, Java Server Programming: Java EE 5 (J2EE 1.5) Black Book, Platinum Edition, is a one-time reference book that covers all aspects of Java EE in an easy-to-understand approach for example, how an application server runs; how GlassFish Application server deploys a Java application; a complete know-how of design patterns, best practices, and design strategies; working with Java related technologies such as NetBeans IDE 6.0, Hibernate, Spring, and Seam frameworks; and proven solutions using the key Java EE technologies, such as JDBC, Servlets, JSP, JSTL, RMI, JNDI, JavaMail, Web services, JCA, Struts, JSF, UML, and much more& All this, as the book explores these concepts with appropriate examples and executable applications no doubt, every aspect of the book is worth its price.
Euro-ParConferenceSeries The European Conference on Parallel Computing (Euro-Par) is an international conference series

dedicated to the promotion and advancement of all aspects of parallel and distributed computing. The major themes fall into the categories of hardware, software, algorithms, and applications. This year, new and interesting topicswereintroduced,likePeer-to-PeerComputing,DistributedMultimedia- stems, and Mobile and Ubiquitous Computing. For the ?rst time, we organized a Demo Session showing many challenging applications. The general objective of Euro-Par is to provide a forum promoting the de- lopment of parallel and distributed computing both as an industrial technique and an academic discipline, extending the frontiers of both the state of the art and the state of the practice. The industrial importance of parallel and dist- buted computing is supported this year by a special Industrial Session as well as a vendors' exhibition. This is particularly important as currently parallel and distributed computing is evolving into a globally important technology; the b- zword Grid Computing clearly expresses this move. In addition, the trend to a - bile world is clearly visible in this year's Euro-Par. ThemainaudienceforandparticipantsatEuro-Parareresearchersinaca- mic departments, industrial organizations, and government laboratories. Euro- Par aims to become the primary choice of such professionals for the presentation of new results in their speci?c areas. Euro-Par has its own Internet domain with a permanent Web site where the history of the conference series is described: http://www.euro-par.org. The Euro-Par conference series is sponsored by the Association for Computer Machinery (ACM) and the International Federation for Information Processing (IFIP).

Introduction to system performance; Monitoring system activity; Managing the workload; Memory performance; Disk performance issues; Network performance; Terminal performance; Kernel configuration.

This book outlines a set of issues that are critical to all of parallel architecture--communication latency, communication bandwidth, and coordination of cooperative work (across modern designs). It describes the set of techniques available in hardware and in software to address each issues and explore how the various techniques interact.

Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively Make informed decisions by identifying the strengths and weaknesses of different tools Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity Understand the distributed systems research upon which modern databases are built Peek behind the scenes of major online services, and learn from their architectures

Your Python code may run correctly, but you need it to run faster. Updated for Python 3, this expanded edition shows you how to locate performance bottlenecks and significantly speed up your code in high-data-volume programs. By exploring the fundamental theory behind design choices, High Performance Python helps you gain a deeper understanding of Python's implementation. How do you take advantage of multicore architectures or clusters? Or build a system that scales up and down without losing reliability? Experienced Python programmers will learn concrete solutions to many issues, along with war stories from companies that use high-performance Python for social media analytics, productionized machine learning, and more. Get a better grasp of NumPy, Cython, and profilers Learn how Python abstracts the underlying computer architecture Use profiling to find bottlenecks in CPU time and memory usage Write efficient programs by choosing appropriate data structures Speed up matrix and vector computations Use tools to compile Python down to machine code Manage multiple I/O and computational operations concurrently Convert multiprocessing code to run on local or remote clusters Deploy code faster using tools like Docker

The Designer's Guide to the Cortex-M Family is a tutorial-based book giving the key concepts required to develop programs in C with a Cortex M- based processor. The book begins with an overview of the Cortex- M family, giving architectural descriptions supported with practical examples, enabling the engineer to easily develop basic C programs to run on the Cortex- M0/M0+/M3 and M4. It then examines the more advanced features of the Cortex architecture such as memory protection, operating modes and dual stack operation. Once a firm grounding in the Cortex M processor has been established the book introduces the use of a small footprint RTOS and the CMSIS DSP library. With this book you will learn: The key differences between the Cortex M0/M0+/M3 and M4 How to write C programs to run on Cortex-M based processors How to make best use of the Coresight debug system How to do RTOS development The Cortex-M operating modes and memory protection Advanced software techniques that can be used on Cortex-M microcontrollers How to optimise DSP code for the cortex M4 and how to build real time DSP systems An Introduction to the Cortex microcontroller software interface standard (CMSIS), a common framework for all Cortex M- based microcontrollers Coverage of the CMSIS DSP library for Cortex M3 and M4 An evaluation tool chain IDE and debugger which allows the accompanying example projects to be run in simulation on the PC or on low cost hardware

Build a scalable, fault-tolerant and highly available data layer for your applications using Apache Cassandra About This Book Install Cassandra and set up multi-node clusters Design rich schemas that capture the relationships between different data types Master the advanced features available in Cassandra 3.x through a step-by-step tutorial and build a scalable, high performance database layer Who This Book Is For If you are a NoSQL developer and new to Apache Cassandra who wants to learn its common as well as not-so-common features, this book is for you. Alternatively, a developer wanting to enter the world of NoSQL will find this book useful. It does not assume any prior experience in coding or any framework. What You Will Learn Install Cassandra Create keyspaces and tables with multiple clustering columns to organize related data Use secondary indexes and materialized views to avoid denormalization of data Effortlessly handle concurrent updates with collection columns Ensure data integrity with lightweight transactions and logged batches Understand eventual consistency and use the right consistency level for your situation Understand data distribution with Cassandra Develop simple application using Java driver and implement application-level optimizations In Detail Cassandra is a distributed database that stands out thanks to its robust feature set and intuitive interface, while providing high availability and scalability of a distributed data store. This book will introduce you to the rich feature set offered by Cassandra, and empower you to create and manage a highly scalable, performant and fault-tolerant database layer. The book starts by explaining the new features implemented in Cassandra 3.x and get you set up with Cassandra. Then you'll walk through data modeling in Cassandra and the rich feature set available to design a flexible schema. Next you'll learn to create tables with composite partition keys, collections and user-defined types and get to know different methods to avoid

denormalization of data. You will then proceed to create user-defined functions and aggregates in Cassandra. Then, you will set up a multi node cluster and see how the dynamics of Cassandra change with it. Finally, you will implement some application-level optimizations using a Java client. By the end of this book, you'll be fully equipped to build powerful, scalable Cassandra database layers for your applications. Style and approach This book takes a step-by- step approach to give you basic to intermediate knowledge of Apache Cassandra. Every concept is explained in depth, and is supplemented with practical examples when required.