

Prefácio

*O que podemos experimentar de mais belo é o mistério.
Ele é a fonte de toda arte e ciência verdadeira.*

Albert Einstein, Como vejo o mundo, 1930

Este livro

Acreditamos que o aprendizado na Ciência da Computação e na Engenharia deve refletir o estado atual da área, além de apresentar os princípios que estão moldando a computação. Também achamos que os leitores em cada especialidade da computação precisam apreciar os paradigmas organizacionais que determinam as capacidades, o desempenho e, por fim, o sucesso dos sistemas computacionais.

A tecnologia computacional moderna exige que os profissionais de cada especialidade da computação entendam tanto o hardware quanto o software. A interação entre hardware e software em diversos níveis também oferece uma estrutura para se entender os fundamentos da computação. Não importa se seu interesse principal é hardware ou software, Ciência da Computação ou Engenharia Elétrica, as ideias centrais na organização e projeto de computadores são as mesmas. Assim, nossa ênfase neste livro é mostrar o relacionamento entre hardware e software e apresentar os conceitos que são a base para os computadores atuais.

A passagem recente de processador para microprocessadores multicore confirmou a solidez desse ponto de vista, dado desde a primeira edição. Embora os programadores pudessem ignorar e confiar em arquitetos de computador, escritores de compilador e engenheiros de silício para fazer os seus programas executarem mais rápido, sem mudanças, essa era já terminou. Para os programas executarem mais rápido, eles precisam se tornar paralelos. Embora o objetivo de muitos pesquisadores seja possibilitar que os programadores não precisem saber a natureza paralela subjacente do hardware que eles estão programando, serão necessários muitos anos para se concretizar essa visão. Nossa visão é que, pelo menos na próxima década, a maioria dos programadores terá de entender a interface hardware/software se quiser que os programas executem de modo eficiente em computadores paralelos.

Este livro é útil para aqueles com pouca experiência em linguagem assembly ou projeto lógico, que precisam entender a organização básica do computador, e também para leitores com base em linguagem assembly e/ou projeto lógico, que queiram aprender a projetar um computador ou entender como um sistema funciona e por que se comporta de determinada forma.

O outro livro

Alguns leitores podem estar familiarizados com *Arquitetura de Computadores: Uma abordagem quantitativa*, conhecido popularmente como Hennessy e Patterson. (Este livro, por sua vez, é chamado Patterson e Hennessy.) Nossa motivação ao escrever aquele livro foi descrever os princípios da arquitetura de computadores usando fundamentos sólidos de engenharia e a relação custo/benefício. Usamos um enfoque que combinava exemplos e

medições, baseado em sistemas comerciais, para criar experiências de projeto realísticas. Nosso objetivo foi demonstrar que arquitetura de computadores poderia ser aprendida por meio de metodologias quantitativas, em vez de por uma técnica descritiva. O livro era voltado para profissionais de computação sérios, que desejam um conhecimento detalhado dos computadores.









































































A maioria dos leitores deste livro não planeja se tornar arquiteto de computador. Contudo, o desempenho dos sistemas de software futuros será drasticamente afetado pela forma como os projetistas de software entendem as técnicas de hardware básicas em funcionamento em um sistema. Assim, aqueles que escrevem compiladores, projetistas de sistema operacional, programadores de banco de dados e a maioria dos outros engenheiros de software precisam de um fundamento sólido sobre os princípios apresentados neste livro. De modo semelhante, os projetistas de hardware precisam entender claramente os efeitos de seu trabalho sobre as aplicações de software.

Assim, sabíamos que este livro tinha de ser muito mais do que um subconjunto do material contido em *Arquitetura de Computadores*, e o material foi bastante revisado para corresponder a esse público-alvo diferente. Ficamos tão satisfeitos com o resultado que as edições seguintes do *Arquitetura de Computadores* foram revisadas para remover a maior parte do material introdutório; logo, há muito menos repetição hoje do que nas primeiras edições dos dois livros.

Mudanças para a quarta edição

Tivemos cinco objetivos principais para esta terceira edição de *Organização e Projeto de Computadores*: dada a revolução *multicore* nos microprocessadores, destacar os tópicos de hardware e software paralelo no decorrer do livro; dinamizar o conteúdo existente de modo a dar espaço para tópicos sobre paralelismo; melhorar a pedagogia em geral; atualizar o conteúdo técnico para refletir mudanças ocorridas na área desde a publicação da terceira edição em 2004; e restaurar a utilidade dos exercícios nesta era da Internet.

Antes de discutirmos os objetivos com detalhes, vejamos a tabela a seguir. Ela mostra as sequências de hardware e software no decorrer do livro. Os Capítulos 1, 4, 5 e 7 são encontrados nas duas sequências, não importando a experiência ou o foco. O Capítulo 1 é uma nova introdução, que inclui uma discussão sobre a importância da potência e como ela motiva a mudança de microprocessadores de “core” único para “multicore”. Também apresenta conteúdo sobre desempenho e benchmarking, que tinham um capítulo separado na terceira edição. O Capítulo 2 provavelmente será material de revisão para os que são voltados para o hardware, mas é uma leitura essencial para aqueles voltados para o software, especialmente para os leitores interessados em aprender mais sobre os compiladores e as linguagens de programação orientadas a objeto. Ele inclui material do Capítulo 3 da terceira edição, de modo que a arquitetura MIPS completa agora está em um único capítulo, menos as instruções de ponto flutuante. O Capítulo 3 é para os leitores interessados em construir um caminho de dados ou em aprender mais sobre aritmética de ponto flutuante. Alguns pularão o Capítulo 3, ou porque não precisam dele ou porque é uma revisão. O Capítulo 4 combina dois capítulos da terceira edição para explicar os processadores em pipeline. As Seções 4.1, 4.5 e 4.10 oferecem resumos para os que são voltados a software. Porém, aqueles mais interessados em hardware descobrirão que esse capítulo apresenta um material básico; eles também podem, dependendo de sua base, querer ler primeiro o Apêndice C, sobre projeto lógico. O Capítulo 6, que trata de armazenamento, é essencial para os leitores com foco no software, e deve ser lido pelos outros, se houver tempo. O último capítulo sobre multicore, multiprocessadores e clusters é um material basicamente novo, e deve ser lido por todos.


Capítulo ou apêndice	Seções	Foco no software	Foco no hardware
1. Abstrações e Tecnologias Computacionais	1.1 a 1.9		
	 1.10 (História)		
2. Instruções: A Linguagem de Máquina	2.1 a 2.14		
	 2.15 (Compiladores & Java)		
	2.16 a 2.19		
	 2.20 (História)		
E. Arquiteturas do conjunto de instruções RISC	 E.1 a E.19		
3. Aritmética Computacional	3.1 a 3.9		
	 3.10 (História)		
C. Fundamentos do Projeto Lógico	 C.1 a C.13		
4. O Processador	4.1 (Visão Geral)		
	4.2 (Convenções Lógicas)		
	4.3 a 4.4 (Implementação Simples)		
	4.5 (Visão Geral do Pipelining)		
	4.6 (Caminho de Dados em Pipeline)		
	4.7 a 4.9 (Hazards, Exceções)		
	4.10 a 4.11 (Paralelo, Vida Real)		
	 4.12 (Controle de Pipeline Verilog)		
	4.13 a 4.14 (Falácias)		
	 4.15 (História)		
D. Mapeando o Controle no Hardware	 D.1 a D.6		
5. Grande e Rápida: Explorando a Hierarquia de Memória	5.1 a 5.8		
	 5.9 (Controlador de Cache Verilog)		
	5.10 a 5.12		
	 5.13 (História)		
6. Armazenamento e outros tópicos de E/S	6.1 a 6.10		
	 6.11 (Redes)		
	6.12 a 6.13		
	 6.14 (História)		
7. Multicores, multiprocessadores e clusters	7.1 a 7.13		
	 7.14 (História)		
A. Unidades de Processamento Gráfico	A.1 a A.12		
B. Montadores, Link-editores e o Simulador SPIM	B.1 a B.12		

Leia cuidadosamente 

Leia se tiver tempo 

Referência 

Revise ou leia 

Leia para ter cultura 

O primeiro objetivo foi tornar o paralelismo um cidadão de primeira classe nesta edição, pois era um capítulo separado no material complementar da edição anterior. O exemplo mais óbvio é o Capítulo 7. Em particular, esse capítulo apresenta o modelo de desempenho Roofline, e mostra seu valor, avaliando quatro arquiteturas multicore recentes em dois kernels. Esse modelo poderia demonstrar ser tão compreensível para multiprocessadores multicore quanto o modelo 3Cs é para os caches.

Dada a importância do paralelismo, não seria sensato esperar até o último capítulo para falar a respeito, de modo que existe uma seção sobre paralelismo em cada um dos seis capítulos anteriores:

- [B] *Capítulo 1: Paralelismo e Potência*. Mostra como os limites de potência forçaram a indústria a passar para o paralelismo, e por que o paralelismo ajuda nesse aspecto.
- [B] *Capítulo 2: Paralelismo e Instruções: Sincronização*. Esta seção discute os bloqueios para variáveis compartilhadas, especificamente as instruções Load Linked e Store Conditional do MIPS.
- [B] *Capítulo 3: Paralelismo e Aritmética de Computador: Associatividade de Ponto Flutuante*. Essa seção discute os desafios da precisão numérica e cálculos de ponto flutuante.
- [B] *Capítulo 4: Paralelismo e Paralelismo Avançado em Nível de Instrução*. Aborda ILP avançado — superescalar, especulação, VLIW, desdobramento de loop e OOO — além do relacionamento entre profundidade de pipeline e consumo de potência.
- [B] *Capítulo 5: Paralelismo e Hierarquias de Memória: Coerência de Cache*. Apresenta coerência, consistência e snooping dos protocolos de cache.
- [B] *Capítulo 6: Paralelismo e E/S: Redundant Arrays of Inexpensive Disks*. Descreve RAID como um sistema de E/S paralela, além do sistema ICO altamente disponível.

O Capítulo 7 conclui com os motivos para o otimismo pelo qual a incursão no paralelismo deverá ter mais sucesso do que aquelas do passado.

Estou particularmente entusiasmado com o acréscimo de um apêndice sobre Unidades de Processamento Gráfico, escrito pelo cientista chefe da NVIDIA, David Kirk, e arquiteto chefe John Nickolls. O Apêndice A é a primeira descrição em profundidade das GPUs, que representam uma investida nova e interessante em arquitetura de computador. O apêndice se baseia em temas paralelos dessa edição para apresentar um estilo de computação que permite que o programador pense em MIMD embora o hardware tente executar no estilo SIMD sempre que for possível. Como as GPUs são baratas e encontradas com facilidade — até mesmo em muitos laptops — e seus ambientes de programação estão disponíveis gratuitamente, elas oferecem uma plataforma de hardware paralela que muitos poderiam experimentar.

O segundo objetivo foi enxugar o livro para criar espaço para o material novo sobre paralelismo. O primeiro passo foi simplesmente passar um pente fino em todos os parágrafos acumulados nas três edições anteriores, para ver se ainda eram necessários. As principais mudanças foram a junção de capítulos e o descarte de tópicos. Mark Hill sugeriu a remoção da implementação de processador multiciclo e, em vez disso, acrescentar um controlador de cache multiciclo ao capítulo sobre hierarquia de memória. Isso permitiu que o processador fosse apresentado em um único capítulo, em vez de dois, melhorando o material sobre processador pela omissão. O material sobre desempenho, de um capítulo separado na terceira edição, agora está junto com o primeiro capítulo.

O terceiro objetivo foi melhorar a pedagogia do livro. O Capítulo 1 agora está mais recheado, incluindo desempenho, circuitos integrados e potência, e prepara o palco para o

restante do livro. Os Capítulos 2 e 3 foram escritos originalmente em um estilo evolutivo, começando com uma arquitetura de “célula única” e terminando com a arquitetura MIPS completa ao final do Capítulo 3. Esse estilo vagaroso não combina com o leitor moderno. Esta edição junta todo o material sobre conjunto de instruções para as instruções de inteiros no Capítulo 2 — tornando o Capítulo 3 opcional para muitos leitores — e cada seção agora tem significado próprio. O leitor não precisa mais ler todas as seções anteriores. Logo, o Capítulo 2 agora é ainda melhor como referência do que nas edições anteriores. O Capítulo 4 funciona melhor porque o processador agora está em um único capítulo, pois a implementação de multiciclos hoje é uma distração. O Capítulo 5 tem uma nova seção sobre a montagem de controladores de cache.

O Material Complementar nos permitiu reduzir o custo do livro, economizando páginas e também aprofundando-nos em assuntos que eram de interesse de alguns, mas não de todos os leitores. Cada capítulo agora tem a seção de Perspectivas Históricas no site (www.elsevier.com.br/organizacaoeprojeto), e quatro capítulos também possuem uma seção de conteúdo avançado.

Para aqueles que perguntam por que incluímos material complementar além do livro, a resposta é simples: o material complementar apresenta conteúdo que acreditamos que será fácil e imediatamente acessível ao leitor, não importa onde ele esteja. Se você estiver interessado no conteúdo avançado, ou se quiser rever um tutorial sobre VHDL (por exemplo), ele está no site, pronto para você utilizar.

Este é um campo de mudanças rápidas, e como sempre acontece para nossas novas edições, um objetivo importante é atualizar o conteúdo técnico. O AMD Opteron X4 modelo 2356 (apelidado de “Barcelona”) serve como um exemplo contínuo no livro, e aparece nos Capítulos 1, 4, 5 e 7. Os Capítulos 1 e 6 acrescentam resultados do novo benchmark de potência do SPEC. O Capítulo 2 acrescenta uma seção sobre arquitetura ARM, que atualmente é a ISA de 32 bits mais comum. O Capítulo 5 acrescenta uma nova seção sobre máquinas virtuais, que estão ressurgindo em importância. O Capítulo 5 possui medições detalhadas de desempenho de cache no multicore Opteron X4 e alguns detalhes sobre seu rival, o Intel Nehalem, que foi anunciado no final de 2008. O Capítulo 6 descreve a memória flash pela primeira vez, além de um servidor incrivelmente compacto da Sun, que comprime 8 núcleos, 16 DIMMs e 8 discos em um único bit 1U. Ele também inclui os resultados recentes sobre falhas de disco a longo prazo. O Capítulo 7 aborda diversos tópicos com relação a paralelismo — incluindo multithreading, SIMD, vetor, GPUs, modelos de desempenho, benchmarks, redes de multiprocessadores — e descreve três multicores mais o Opteron X4: Intel Xeon modelo e5345 (Clovertown), IBM Cell modelo QS20 e o Sun Microsystems T2 modelo 5120 (Niagara 2).

O objetivo final foi tentar tornar os exercícios úteis a instrutores nesta era da Internet, pois os trabalhos de casa há muito tempo têm sido um modo importante de aprender sobre o material. Infelizmente, hoje as respostas são postadas on-line assim que o livro aparece. Temos uma técnica em duas partes. Primeiro, colaboradores especialistas trabalharam para desenvolver exercícios inteiramente novos para cada capítulo do livro. Segundo, a maioria dos exercícios possui uma descrição qualitativa com o apoio de uma tabela que oferece vários parâmetros quantitativos alternativos, necessários para responder a essa questão. O simples número mais a flexibilidade em termos de como o instrutor pode escolher para atribuir variações dos exercícios tornará difícil para os alunos encontrarem as soluções correspondentes on-line. Os instrutores também poderão mudar esses parâmetros quantitativos como desejarem, novamente frustrando os alunos que contam com a Internet para buscar soluções para um conjunto de exercícios estático e inalterável. Achamos que essa nova técnica é um novo acréscimo valioso ao livro — por favor, nos informe a sua opinião, seja como aluno ou como instrutor!

Preservamos elementos úteis do livro das edições anteriores. Para fazer com que o livro funcione melhor como referência, ainda colocamos definições de novos termos nas margens, em sua primeira ocorrência. A seção “Entendendo o Desempenho do Programa”, presente em cada capítulo, ajuda os leitores a entenderem o desempenho de seus programas e como melhorá-lo, assim como o elemento “Interface de Hardware/Software” do livro ajudou os leitores a entenderem as escolhas nessa interface. A seção “Colocando em Perspectiva” continua, de modo que o leitor verá a floresta apesar de todas as árvores. As seções “Verifique Você Mesmo” ajudam os leitores a confirmarem sua compreensão do material na primeira vez com as respostas fornecidas ao final de cada capítulo. Esta edição também inclui a placa de referência MIPS verde, que foi inspirada pelo “Green Card” do IBM System/360. O cartão removível foi atualizado e deverá ser uma referência prática na escrita de programas em linguagem assembly MIPS.

Comentários finais

Ao ler a seção de agradecimentos a seguir, você verá que trabalhamos bastante para corrigir erros. Como um livro passa por muitas tiragens, temos a oportunidade de fazer ainda mais correções. Se você descobrir quaisquer outros erros, por favor, entre em contato com a editora por correio eletrônico em atendimento1@elsevier.com, ou pelo correio *low-tech*, usando o endereço encontrado na página de copyright.

Esta edição marca uma quebra na colaboração duradoura entre Hennessy e Patterson, que começou em 1989. As demandas da condução de uma das maiores universidades do mundo significam que o Presidente Hennessy não poderia mais se comprometer com a escrita de uma nova edição. O outro autor se sentiu como um malabarista que sempre trabalhou com um parceiro e que, de repente, é lançado no palco como um ato solo. Assim, as pessoas nos agradecimentos e os colegas da Berkeley desempenharam um papel ainda maior na formação do conteúdo deste livro. Apesar disso, desta vez só existe um único autor a culpar pelo novo material que você está para ler.

Agradecimentos da quarta edição

Gostaria de agradecer a **David Kirk**, **John Nickolls** e seus colegas na NVIDIA (Michael Garland, John Montrym, Doug Voorhies, Lars Nyland, Erik Lindholm, Paulius Micikevicius, Massimiliano Fatica, Stuart Oberman e Vasily Volkov) por escreverem o primeiro apêndice detalhado sobre GPUs. Gostaria de expressar novamente meu apreço a **Jim Larus**, da Microsoft Research, por sua disposição em contribuir com sua experiência em programação na linguagem assembly, além de aceitar que os leitores deste livro usem o simulador que ele desenvolveu e mantém.

Também sou agradecido pelas contribuições de muitos especialistas que desenvolveram os novos exercícios para esta nova edição. A escrita de bons exercícios não é uma tarefa fácil, e cada colaborador trabalhou muito para desenvolver problemas que sejam desafiadores e atraentes:

- *Capítulo 1:* **Javier Bruguera** (Universidade de Santiago de Compostela)
- *Capítulo 2:* **John Oliver** (Cal Poly, San Luis Obispo), com colaborações de **Nicole Kaiyan** (University of Adelaide) e **Milos Prvulovic** (Georgia Tech)
- *Capítulo 3:* **Matthew Farrens** (University of California, Davis)
- *Capítulo 4:* **Milos Prvulovic** (Georgia Tech)

- **Capítulo 5: Jichuan Chang, Jacob Leverich, Kevin Lim e Partha Ranganathan** (todos da Hewlett-Packard), com colaborações de Nicole Kaiyan (University of Adelaide)
- **Capítulo 6: Perry Alexander** (The University of Kansas)
- **Capítulo 7: David Kaeli** (Northeastern University)

Peter Ashenden realizou um esforço gigantesco de edição e avaliação de *todos* os novos exercícios.

Obrigado a **David August e Prakash Prabhu**, da Princeton University, por seu trabalho nos exames de capítulo que estão disponíveis para os instrutores no site da editora.

Contei com meus colegas do Vale do Silício para grande parte do material que este livro utiliza:

- **AMD** — para os detalhes e medição do Opteron X4 (Barcelona): **William Brantley, Vasileios Liaskovitis, Chuck Moore e Brian Waldecker**.
- **Intel** — para a informação de pré-lançamento sobre o Intel Nehalem: **Faye Briggs**.
- **Micron** — para a base sobre Memória Flash no Capítulo 6: **Dean Klein**.
- **Sun Microsystems** — para as medições dos mixes de instruções nos benchmarks SPEC2006 no Capítulo 2 e detalhes e medições do Sun Server x4150 no Capítulo 6: **Yan Fisher, John Fowler, Darryl Gove, Paul Joyce, Shenik Mehta, Pierre Reynes, Dimitry Stuve, Durgam Vahia e David Weaver**.
- **U.C. Berkeley** — **Krste Asanovic** (que forneceu a ideia para concorrência de software *versus* paralelismo do hardware no Capítulo 7), **James Demmel e Velvel Kahan** (que comentaram sobre paralelismo e cálculos de ponto flutuante), **Zhangxi Tan** (que projetou o controlador de cache e escreveu o Verilog para ele no Capítulo 5), **Sam Williams** (que forneceu o modelo Roofline e as medições multicore no Capítulo 7), e o restante dos meus colegas na **Par Lab**, que me deram muitas sugestões e opiniões sobre os tópicos de paralelismo encontrados no decorrer do livro.

Sou grato aos muitos instrutores que responderam às pesquisas da editora, revisaram nossas propostas e participaram de grupos de foco para analisar e responder aos nossos planos para esta edição. Entre eles estão os seguintes: *Grupo de foco*: Mark Hill (University of Wisconsin, Madison), E. J. Kim (Texas A&M University), Jihong Kim (Seoul National University), Lu Peng (Louisiana State University), Dean Tullsen (UC San Diego), Ken Vollmar (Missouri State University), David Wood (University of Wisconsin, Madison), Ki Hwan Yum (University of Texas, San Antonio); *Inspeções e críticas*: Mahmoud Abou-Nasr (Wayne State University), Perry Alexander (The University of Kansas), Hakan Aydin (George Mason University), Hussein Badr (State University of New York em Stony Brook), Mac Baker (Virginia Military Institute), Ron Barnes (George Mason University), Douglas Blough (Georgia Institute of Technology), Kevin Bolding (Seattle Pacific University), Miodrag Bolic (University of Ottawa), John Bonomo (Westminster College), Jeff Braun (Montana Tech), Tom Briggs (Shippensburg University), Scott Burgess (Humboldt State University), Fazli Can (Bilkent University), Warren R. Carithers (Rochester Institute of Technology), Bruce Carlton (Mesa Community College), Nicholas Carter (University of Illinois at Urbana-Champaign), Anthony Cocchi (The City University of New York), Don Cooley (Utah State University), Robert D. Cupper (Allegheny College), Edward W. Davis (North Carolina State University), Nathaniel J. Davis (Air Force Institute of Technology), Molisa Derk (Oklahoma City University), Derek Eager (University of Saskatchewan), Ernest Ferguson (Northwest Missouri State University), Rhonda Kay Gaede (The University of Alabama), Etienne M. Gagnon (UQAM), Costa Gerousis

(Christopher Newport University), Paul Gillard (Memorial University of Newfoundland), Michael Goldweber (Xavier University), Georgia Grant (College of San Mateo), Merrill Hall (The Master's College), Tyson Hall (Southern Adventist University), Ed Harcourt (Lawrence University), Justin E. Harlow (University of South Florida), Paul F. Hemler (Hampden-Sydney College), Martin Herbordt (Boston University), Steve J. Hodges (Cabrillo College), Kenneth Hopkinson (Cornell University), Dalton Hunkins (St. Bonaventure University), Baback Izadi (State University of New York — New Paltz), Reza Jafari, Robert W. Johnson (Colorado Technical University), Bharat Joshi (University of North Carolina, Charlotte), Nagarajan Kandasamy (Drexel University), Rajiv Kapadia, Ryan Kastner (University of California, Santa Barbara), Jim Kirk (Union University), Geoffrey S. Knauth (Lycoming College), Manish M. Kochhal (Wayne State), Suzan Koknar-Tezel (Saint Joseph's University), Angkul Kongmunvattana (Columbus State University), April Kontostathis (Ursinus College), Christos Kozyrakis (Stanford University), Danny Krizanc (Wesleyan University), Ashok Kumar, S. Kumar (The University of Texas), Robert N. Lea (University of Houston), Baoxin Li (Arizona State University), Li Liao (University of Delaware), Gary Livingston (University of Massachusetts), Michael Lyle, Douglas W. Lynn (Oregon Institute of Technology), Yashwant K Malaiya (Colorado State University), Bill Mark (University of Texas at Austin), Ananda Mondal (Claflin University), Alvin Moser (Seattle University), Walid Najjar (University of California, Riverside), Danial J. Neebel (Loras College), John Nestor (Lafayette College), Joe Oldham (Centre College), Timour Paltashev, James Parkerson (University of Arkansas), Shaunak Pawagi (SUNY em Stony Brook), Steve Pearce, Ted Pedersen (University of Minnesota), Gregory D Peterson (The University of Tennessee), Dejan Raskovic (University of Alaska, Fairbanks) Brad Richards (University of Puget Sound), Roman Rozanov, Louis Rubinfeld (Villanova University), Md Abdus Salam (Southern University), Augustine Samba (Kent State University), Robert Schaefer (Daniel Webster College), Carolyn J. C. Schauble (Colorado State University), Keith Schubert (CSU San Bernardino), William L. Schultz, Kelly Shaw (University of Richmond), Shahram Shirani (McMaster University), Scott Sigman (Drury University), Bruce Smith, David Smith, Jeff W. Smith (University of Georgia, Athens), Philip Snyder (Johns Hopkins University), Alex Sprintson (Texas A&M), Timothy D. Stanley (Brigham Young University), Dean Stevens (Morningside College), Nozar Tabrizi (Kettering University), Yuval Tamir (UCLA), Alexander Taubin (Boston University), Will Thacker (Winthrop University), Mithuna Thottethodi (Purdue University), Manghui Tu (Southern Utah University), Rama Viswanathan (Beloit College), Guoping Wang (Indiana-Purdue University), Patricia Wenner (Bucknell University), Kent Wilken (University of California, Davis), David Wolfe (Gustavus Adolphus College), David Wood (University of Wisconsin, Madison), Mohamed Zahran (City College of New York), Gerald D. Zarnett (Ryerson University), Nian Zhang (South Dakota School of Mines & Technology), Jiling Zhong (Troy University), Huiyang Zhou (The University of Central Florida), Weiyu Zhu (Illinois Wesleyan University).

Gostaria de agradecer especialmente ao pessoal da Berkeley, que deu a opinião principal para o Capítulo 7 e o Apêndice A, que foram as partes mais desafiadoras para se escrever nesta edição: **Krste Asanovic, Christopher Batten, Rastilav Bodik, Bryan Catanzaro, Jake Chong, Kaushik Data, Greg Giebling, Anik Jain, Jae Lee, Vasily Volkov e Samuel Williams.**

Um agradecimento especial a **Mark Smotherman**, que fez uma revisão final cuidadosa, para descobrir problemas técnicos e de escrita, o que melhorou significativamente a qualidade desta edição. Ele desempenhou um papel ainda mais importante neste ato, visto que esta edição foi feita como um ato solo.

Gostaríamos de agradecer a toda a família Morgan Kaufmann, que concordou em publicar este livro novamente, sob a liderança capaz de **Denise Penrose. Nathaniel McFadden** foi revisor de desenvolvimento para esta edição, e trabalhou comigo semanalmente no conteúdo do livro. **Kimberlee Honjo** coordenou a inspeção de usuários e suas respostas.

Dawnmarie Simpson administrou o processo de produção do livro. Também agradecemos aos muitos fornecedores autônomos que contribuíram para este volume, especialmente Alan Rose, da Multiscience Press, e à diacriTech, pela composição.

As contribuições de quase 200 pessoas que mencionamos aqui tornaram esta quarta edição nosso melhor livro até agora. Divirta-se!

David A. Patterson