Arquitetura de Computadores:

Uma Abordagem Quantitativa

Quinta Edição

Arquitetura de Computadores:

Uma Abordagem Quantitativa

Quinta Edição

John L. Hennessy
Universidade de Stanford

David A. Patterson
Universidade da Califórnia, Berkeley

Com contribuições de

Krste Asanovic
Universidade da Califórnia, Berkeley

Jason D. Bakos
Universidade da Carolina do Sul

Robert P. Colwell
R&E Colwell & Assoc. Inc.

Thomas M. Conte
Universidade do Estado da Carolina do Norte

José Duato
Universidade Politécnica de Valência e Simula

Diana Franklin
Universidade da Califórnia, Santa Bárbara

David Goldberg
Instituto de Pesquisa Scripps

Norman P. Jouppi
HP Labs

Sheng Li
HP Labs

Naveen Muralimanohar
HP Labs

Gregory D. Peterson
Universidade do Tennessee

Timothy M. Pinkston
Universidade do Sul da Califórnia

Parthasarathy Ranganathan
HP Labs

David A. Wood
Universidade do Wisconsin–Madison

Amr Zaky
Universidade de Santa Clara

ELSEVIER

Amsterdã • Boston • Heidelberg • Londres •
Nova York • Oxford • Paris • São Diego •
São Francisco • Cingapura • Sydney • Tóquio

MORGAN KAUFMANN

Do original: *Computer Architecture: A Quantitative Approach*
Tradução autorizada do idioma inglês da edição publicada por Morgan Kaufmann, an imprint of Elsevier, Inc.

Copidesque: Andréa Vidal
Revisão Gráfica: Adriana Maria Patrício Takaki / Marco Antonio Corrêa / Roberto Mauro dos Santos Facce*:*
Editoração Eletrônica: Thomson Digital

Elsevier Editora Ltda.
Conhecimento sem Fronteiras
Rua Sete de Setembro, 111 – 16º andar
20050-006 – Centro - Rio de Janeiro – RJ - Brasil

Rua Quintana, 753/8º andar
04569-011 Brooklin - São Paulo - SP - Brasil

Serviço de Atendimento ao Cliente
O800-0265340
Atendimento1@elsevier.com

ISBN: 978-85-352-6122-6
ISBN (versão digital): 978-85-352-6411-1
Edição original: ISBN 978-0-12-383872-8

Nota: Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação ao nosso Serviço de Atendimento ao Cliente, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, ou bens, originados do uso desta publicação.

CIP-BRASIL. CATALOGAÇÃO NA PUBLICAÇÃO
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

H436a
Hennessy, John L.
Arquitetura de computadores : uma abordagem quantitativa / John L. Hennessy, David A. Petterson ; tradução Eduardo Kraszczuk. - [5. ed.] - Rio de Janeiro : Elsevier, 2014.
744 p. : il. ; 28 cm.

Tradução de: Computer architecture, 5th ed. : a quantitative approach
Inclui apêndice
ISBN 978-85-352-6122-6

1. Arquitetura de computador. I. Patterson, David A. II. Título.

13-05666 CDD: 004.22
CDU: 004.2

Sobre os Autores

John L. Hennessy é o décimo presidente da Universidade de Stanford, onde é membro do corpo docente desde 1977, nos departamentos de Engenharia Elétrica e Ciência da Computação. Hennessy é membro do IEEE e ACM, membro da Academia Nacional de Engenharia e da Sociedade Americana de Filosofia e membro da Academia Americana de Artes e Ciências. Entre seus muitos prêmios estão o Prêmio Eckert-Mauchly de 2001, por suas contribuições para a tecnologia RISC, o Prêmio Seymour Cray de Engenharia da Computação de 2001 e o Prêmio John von Neumann de 2000, que ele dividiu com David Patterson. Ele também recebeu sete doutorados honorários.

Em 1981, ele iniciou o Projeto MIPS, em Stanford, com um grupo de estudantes de pós-graduação. Depois de completar o projeto em 1984, tirou licença da universidade para co-fundar a MIPS Computer Systems (hoje MIPS Technologies), que desenvolveu um dos primeiros microprocessadores RISC comerciais. Em 2006, mais de 2 bilhões de microprocessadores MIPS foram vendidos em dispositivos, variando de video games e computadores palmtop a impressoras laser e switches de rede. Em seguida, Hennessy liderou o projeto DASH (Director Architeture for Shared Memory – Arquitetura Diretora para Memória Compartilhada), que criou o protótipo do primeiro microprocessador com cache coerente escalável. Muitas das ideias-chave desse projeto foram adotadas em multiprocessadores modernos. Além de suas atividades técnicas e responsabilidades na universidade, ele continuou a trabalhar com diversas empresas startup como conselheiro nos estágios iniciais e como investidor.

David A. Patterson ensina arquitetura de computadores na Universidade da Califórnia, em Berkeley, desde que se juntou ao corpo docente em 1977, onde ele ocupa a Cadeira Pardee de Ciência da Computação. Sua docência foi honrada com o Prêmio de Ensino Notável da Universidade da Califórnia, o Prêmio Karlstrom da ACM, a Medalha Mulligan de Educação e o Prêmio de Ensino Universitário do IEEE. Patterson recebeu o Prêmio de Realização Técnica do IEEE e o Prêmio Eckert-Mauchly por contribuições para o RISC e dividiu o Prêmio Johnson de Armazenamento de Informações por contribuições para o RAID. Ele também dividiu a Medalha John von Neumann do IEEE e o Prêmio C&C com John Hennessy. Como seu coautor, Patterson é membro da Academia Americana de Artes e Ciências, do Museu da História dos Computadores, ACM e IEEE, e foi eleito para a Academia Nacional de Engenharia, Academia Nacional de Ciências e para o Hall da Fama da Engenharia do Vale do Silício. Ele atuou no Comitê Consultivo de Tecnologia da Informação do presidente dos Estados Unidos, como presidente da divisão de CS no departamento EECS em Berkeley, como presidente da Associação de Pesquisa em Computação e como Presidente da ACM. Este histórico levou a prêmios de Serviço Destacado da ACM e CRA.

Em Berkeley, Patterson liderou o projeto e a implementação do RISC I, provavelmente o primeiro computador com conjunto reduzido de instruções VLSI, e a fundação da arquitetura comercial SPARC. Ele foi líder do projeto Arrays Redundantes de Discos Baratos (Redundant Array of Inexpensive Disks – RAID), que levou a sistemas de armazenamento

confiáveis para muitas empresas. Ele também se envolveu no projeto Rede de Workstations (Network of Workstations – NOW), que levou à tecnologia de clusters usada pelas empresas de Internet e, mais tarde, à computação em nuvem. Esses projetos valeram três prêmios de dissertação da ACM. Seus projetos de pesquisa atuais são o Laboratório Algoritmo-Máquina-Pessoas e o Laboratório de Computação Paralela, onde ele é o diretor. O objetivo do Laboratório AMP é desenvolver algoritmos de aprendizado de máquina escaláveis, modelos de programação amigáveis para computadores em escala de depósito e ferramentas de crowd-sourcing para obter rapidamente insights valiosos de muitos dados na nuvem. O objetivo do laboratório Par é desenvolver tecnologias para entregar softwares escaláveis, portáveis, eficientes e produtivos para dispositivos pessoais móveis paralelos.

Para Andrea, Linda, e nossos quatro filhos

Elogios para *Arquitetura de Computadores: Uma Abordagem Quantitativa* Quinta Edição

"A 5ª edição de *Arquitetura de Computadores: Uma Abordagem Quantitativa* continua o legado, fornecendo aos estudantes de arquitetura de computadores as informações mais atualizadas sobre as plataformas computacionais atuais e insights arquitetônicos para ajudá-los a projetar sistemas futuros. Um destaque da nova edição é o capítulo significativamente revisado sobre paralelismo em nível de dados, que desmistifica as arquiteturas de GPU com explicações claras, usando terminologia tradicional de arquitetura de computadores."

—Krste Asanovic, Universidade da Califórnia, Berkeley

"*Arquitetura de Computadores: Uma Abordagem Quantitativa* é um clássico que, como um bom vinho, fica cada vez melhor. Eu comprei meu primeiro exemplar quando estava terminando a graduação e ele continua sendo um dos volumes que eu consulto com mais frequência. Quando a quarta edição saiu, havia tanto conteúdo novo que eu precisava comprá-la para continuar atualizado. E, enquanto eu revisava a quinta edição, percebi que Hennessy e Patterson tiveram sucesso de novo. Todo o conteúdo foi bastante atualizado e só o Capítulo 6 já torna esta nova edição uma leitura necessária para aqueles que realmente querem entender a computação em nuvem e em escala de depósito. Somente Hennessy e Patterson têm acesso ao pessoal do Google, Amazon, Microsoft e outros provedores de computação em nuvem e de aplicações em escala de Internet, e não existe melhor cobertura dessa importante área em outro lugar da indústria."

—James Hamilton, Amazon Web Services

"Hennessy e Patterson escreveram a primeira edição deste livro quando os estudantes de pós-graduação construíam computadores com 50.000 transistores. Hoje, computadores em escala de depósito contêm esse mesmo número de servidores, cada qual consistindo de dúzias de processadores independentes e bilhões de transistores. A evolução da arquitetura de computadores tem sido rápida e incansável, mas *Arquitetura de Computadores: Uma Abordagem Quantitativa* acompanhou o processo com cada edição explicando e analisando com precisão as importantes novas ideias que tornam esse campo tão excitante."

—James Larus, Microsoft Research

"Esta nova adição adiciona um soberbo novo capítulo sobre paralelismo em nível de dados em SIMD de vetor e arquiteturas de GPU. Ele explica conceitos-chave de arquitetura no interior das GPUs de mercado de massa, mapeando-os para termos tradicionais e comparando-os com arquiteturas de vetor e SIMD. Ele chega no momento certo e é relevante à mudança generalizada para a computação por GPU paralela. *Arquitetura de Computadores: Uma Abordagem Quantitativa* continua sendo o primeiro a apresentar uma cobertura completa da arquitetura de importantes novos desenvolvimentos!"

—John Nickolls, NVIDIA

"A nova edição deste livro – hoje um clássico – destaca a ascendência do paralelismo explícito (dados, thread, requisição) dedicando um capítulo inteiro a cada tipo. O capítulo sobre paralelismo de dados é particularmente esclarecedor: a comparação e o contraste

entre SIMD de vetor, SIMD em nível de instrução e GPU ultrapassam o jargão associado a cada arquitetura e expõem as similaridades e diferenças entre elas."

—Kunle Olukotun, Universidade de Stanford

"A 5ª edição de *Arquitetura de Computadores: Uma Abordagem Quantitativa* explora os diversos conceitos paralelos e seus respectivos *trade-offs*. Assim como as edições anteriores, esta nova edição cobre as mais recentes tendências tecnológicas. Um destaque é o grande crescimento dos dispositivos pessoais móveis (Personal Mobile Devices – PMD) e da computação em escala de depósito (Warehouse-Scale Computing – WSC), cujo foco mudou para um equilíbrio mais sofisticado entre desempenho e eficiência energética em comparação com o desempenho bruto. Essas tendências estão alimentando nossa demanda por mais capacidade de processamento, que, por sua vez, está nos levando mais longe no caminho paralelo."

—Andrew N. Sloss, Engenheiro consultor, ARM
Autor de *ARM System Developer's Guide*

Agradecimentos

Embora este livro ainda esteja na quinta edição, criamos dez versões diferentes do conteúdo: três versões da primeira edição (alfa, beta e final) e duas versões da segunda, da terceira e da quarta edições (beta e final). Nesse percurso, recebemos a ajuda de centenas de revisores e usuários. Cada um deles ajudou a tornar este livro melhor. Por isso, decidimos fazer uma lista de todas as pessoas que colaboraram em alguma versão deste livro.

COLABORADORES DA QUINTA EDIÇÃO

Assim como nas edições anteriores, este é um esforço comunitário que envolve diversos voluntários. Sem a ajuda deles, esta edição não estaria tão bem acabada.

Revisores

Jason D. Bakos, University of South Carolina; Diana Franklin, The University of California, Santa Barbara; Norman P. Jouppi, HP Labs; Gregory Peterson, University of Tennessee; Parthasarathy Ranganathan, HP Labs; Mark Smotherman, Clemson University; Gurindar Sohi, University of Wisconsin–Madison; Mateo Valero, Universidad Politécnica de Cataluña; Sotirios G. Ziavras, New Jersey Institute of Technology.

Membros do Laboratório Par e Laboratório RAD da University of California–Berkeley, que fizeram frequentes revisões dos Capítulos 1, 4 e 6, moldando a explicação sobre GPUs e WSCs: Krste Asanovic, Michael Armbrust, Scott Beamer, Sarah Bird, Bryan Catanzaro, Jike Chong, Henry Cook, Derrick Coetzee, Randy Katz, Yun-sup Lee, Leo Meyervich, Mark Murphy, Zhangxi Tan, Vasily Volkov e Andrew Waterman.

Painel consultivo

Luiz André Barroso, Google Inc.; Robert P. Colwell, R&E Colwell & Assoc. Inc.; Krisztian Flautner, VP de R&D na ARM Ltd.; Mary Jane Irwin, Penn State; David Kirk, NVIDIA; Grant Martin, cientista-chefe, Tensilica; Gurindar Sohi, University of Wisconsin–Madison; Mateo Valero, Universidad Politécnica de Cataluña.

Apêndices

Krste Asanovic, University of California–Berkeley (Apêndice G); Thomas M. Conte, North Carolina State University (Apêndice E); José Duato, Universitat Politècnica de València and Simula (Apêndice F); David Goldberg, Xerox PARC (Apêndice J); Timothy M. Pinkston, University of Southern California (Apêndice F).

José Flich, da Universidad Politécnica de Valencia, deu contribuições significativas para a atualização do Apêndice F.

Estudos de caso e exercícios

Jason D. Bakos, University of South Carolina (Capítulos 3 e 4); Diana Franklin, University of California, Santa Barbara (Capítulo 1 e Apêndice C); Norman P. Jouppi, HP Labs (Capítulo 2); Naveen Muralimanohar, HP Labs (Capítulo 2); Gregory Peterson, University of Tennessee (Apêndice A); Parthasarathy Ranganathan, HP Labs (Capítulo 6); Amr Zaky, University of Santa Clara (Capítulo 5 e Apêndice B).

Jichuan Chang, Kevin Lim e Justin Meza auxiliaram no desenvolvimento de testes dos estudos de caso e exercícios do Capítulo 6.

Material adicional

John Nickolls, Steve Keckler e Michael Toksvig da NVIDIA (Capítulo 4, NVIDIA GPUs); Victor Lee, Intel (Capítulo 4, comparação do Core i7 e GPU); John Shalf, LBNL (Capítulo 4, arquiteturas recentes de vetor); Sam Williams, LBNL (modelo *roofline* para computadores no Capítulo 4); Steve Blackburn, da Australian National University, e Kathryn McKinley, da University of Texas, em Austin (Desempenho e medições de energia da Intel, no Capítulo 5); Luiz Barroso, Urs Hölzle, Jimmy Clidaris, Bob Felderman e Chris Johnson do Google (Google WSC, no Capítulo 6); James Hamilton, da Amazon Web Services (Distribuição de energia e modelo de custos, no Capítulo 6).

Jason D. Bakos. da University of South Carolina, desenvolveu os novos *slides* de aula para esta edição.

Mais uma vez, nosso agradecimento especial a Mark Smotherman, da Clemson University, que fez a leitura técnica final do nosso manuscrito. Mark encontrou diversos erros e ambiguidades, e, em consequência disso, o livro ficou muito mais limpo.

Este livro não poderia ter sido publicado sem uma editora, é claro. Queremos agradecer a toda a equipe da Morgan Kaufmann/Elsevier por seus esforços e suporte. Pelo trabalho nesta edição, particularmente, queremos agradecer aos nossos editores Nate McFadden e Todd Green, que coordenaram o painel consultivo, o desenvolvimento dos estudos de caso e exercícios, os grupos de foco, as revisões dos manuscritos e a atualização dos apêndices.

Também temos de agradecer à nossa equipe na universidade, Margaret Rowland e Roxana Infante, pelas inúmeras correspondências enviadas e pela "guarda do forte" em Stanford e Berkeley enquanto trabalhávamos no livro.

Nosso agradecimento final vai para nossas esposas, pelo sofrimento causado pelas leituras, trocas de ideias e escrita realizadas cada vez mais cedo todos os dias.

COLABORADORES DAS EDIÇÕES ANTERIORES

Revisores

George Adams, Purdue University; Sarita Adve, University of Illinois, Urbana–Champaign; Jim Archibald, Brigham Young University; Krste Asanovic, Massachusetts Institute of Technology; Jean-Loup Baer, University of Washington; Paul Barr, Northeastern University; Rajendra V. Boppana, University of Texas, San Antonio; Mark Brehob, University of Michigan; Doug Burger, University of Texas, Austin; John Burger, SGI; Michael Butler; Thomas Casavant; Rohit Chandra; Peter Chen, University of Michigan; as turmas de SUNY Stony Brook, Carnegie Mellon, Stanford, Clemson e Wisconsin; Tim Coe, Vitesse Semiconductor; Robert P. Colwell; David Cummings; Bill Dally; David Douglas; José Duato, Universitat Politècnica de València and Simula; Anthony Duben, Southeast Missouri State University; Susan Eggers, University of Washington; Joel Emer; Barry Fagin, Dartmouth; Joel Ferguson, University of California, Santa

Cruz; Carl Feynman; David Filo; Josh Fisher, Hewlett-Packard Laboratories; Rob Fowler, DIKU; Mark Franklin, Washington University (St. Louis); Kourosh Gharachorloo; Nikolas Gloy, Harvard University; David Goldberg, Xerox Palo Alto Research Center; Antonio González, Intel and Universitat Politècnica de Catalunya; James Goodman, University of Wisconsin–Madison; Sudhanva Gurumurthi, University of Virginia; David Harris, Harvey Mudd College; John Heinlein; Mark Heinrich, Stanford; Daniel Helman, University of California, Santa Cruz; Mark D. Hill, University of Wisconsin–Madison; Martin Hopkins, IBM; Jerry Huck, Hewlett-Packard Laboratories; Wen-mei Hwu, University of Illinois at Urbana–Champaign; Mary Jane Irwin, Pennsylvania State University; Truman Joe; Norm Jouppi; David Kaeli, Northeastern University; Roger Kieckhafer, University of Nebraska; Lev G. Kirischian, Ryerson University; Earl Killian; Allan Knies, Purdue University; Don Knuth; Jeff Kuskin, Stanford; James R. Larus, Microsoft Research; Corinna Lee, University of Toronto; Hank Levy; Kai Li, Princeton University; Lori Liebrock, University of Alaska, Fairbanks; Mikko Lipasti, University of Wisconsin–Madison; Gyula A. Mago, University of North Carolina, Chapel Hill; Bryan Martin; Norman Matloff; David Meyer; William Michalson, Worcester Polytechnic Institute; James Mooney; Trevor Mudge, University of Michigan; Ramadass Nagarajan, University of Texas at Austin; David Nagle, Carnegie Mellon University; Todd Narter; Victor Nelson; Vojin Oklobdzija, University of California, Berkeley; Kunle Olukotun, Stanford University; Bob Owens, Pennsylvania State University; Greg Papadapoulous, Sun Microsystems; Joseph Pfeiffer; Keshav Pingali, Cornell University; Timothy M. Pinkston, University of Southern California; Bruno Preiss, University of Waterloo; Steven Przybylski; Jim Quinlan; Andras Radics; Kishore Ramachandran, Georgia Institute of Technology; Joseph Rameh, University of Texas, Austin; Anthony Reeves, Cornell University; Richard Reid, Michigan State University; Steve Reinhardt, University of Michigan; David Rennels, University of California, Los Angeles; Arnold L. Rosenberg, University of Massachusetts, Amherst; Kaushik Roy, Purdue University; Emilio Salgueiro, Unysis; Karthikeyan Sankaralingam, University of Texas at Austin; Peter Schnorf; Margo Seltzer; Behrooz Shirazi, Southern Methodist University; Daniel Siewiorek, Carnegie Mellon University; J. P. Singh, Princeton; Ashok Singhal; Jim Smith, University of Wisconsin–Madison; Mike Smith, Harvard University; Mark Smotherman, Clemson University; Gurindar Sohi, University of Wisconsin–Madison; Arun Somani, University of Washington; Gene Tagliarin, Clemson University; Shyamkumar Thoziyoor, University of Notre Dame; Evan Tick, University of Oregon; Akhilesh Tyagi, University of North Carolina, Chapel Hill; Dan Upton, University of Virginia; Mateo Valero, Universidad Politécnica de Cataluña, Barcelona; Anujan Varma, University of California, Santa Cruz; Thorsten von Eicken, Cornell University; Hank Walker, Texas A&M; Roy Want, Xerox Palo Alto Research Center; David Weaver, Sun Microsystems; Shlomo Weiss, Tel Aviv University; David Wells; Mike Westall, Clemson University; Maurice Wilkes; Eric Williams; Thomas Willis, Purdue University; Malcolm Wing; Larry Wittie, SUNY Stony Brook; Ellen Witte Zegura, Georgia Institute of Technology; Sotirios G. Ziavras, New Jersey Institute of Technology.

Apêndices

O apêndice sobre vetores foi revisado por Krste Asanovic, do Massachusetts Institute of Technology. O apêndice sobre ponto flutuante foi escrito originalmente por David Goldberg, da Xerox PARC.

Exercícios

George Adams, Purdue University; Todd M. Bezenek, University of Wisconsin–Madison (em memória de sua avó, Ethel Eshom); Susan Eggers; Anoop Gupta; David Hayes; Mark Hill; Allan Knies; Ethan L. Miller, University of California, Santa Cruz; Parthasarathy Ranganathan, Compaq Western Research Laboratory; Brandon Schwartz, University of

Wisconsin–Madison; Michael Scott; Dan Siewiorek; Mike Smith; Mark Smotherman; Evan Tick; Thomas Willis

Estudos de caso e exercícios

Andrea C. Arpaci-Dusseau, University of Wisconsin–Madison; Remzi H. Arpaci Dusseau, University of Wisconsin–Madison; Robert P. Colwell, R&E Colwell & Assoc., Inc.; Diana Franklin, California Polytechnic State University, San Luis Obispo; Wen-mei W. Hwu, University of Illinois em Urbana–Champaign; Norman P. Jouppi, HP Labs; John W. Sias, University of Illinois em Urbana–Champaign; David A. Wood, University of Wisconsin–Madison

Agradecimentos especiais

Duane Adams, Defense Advanced Research Projects Agency; Tom Adams; Sarita Adve, University of Illinois, Urbana–Champaign; Anant Agarwal; Dave Albonesi, University of Rochester; Mitch Alsup; Howard Alt; Dave Anderson; Peter Ashenden; David Bailey; Bill Bandy, Defense Advanced Research Projects Agency; Luiz Barroso, Compaq's Western Research Lab; Andy Bechtolsheim; C. Gordon Bell; Fred Berkowitz; John Best, IBM; Dileep Bhandarkar; Jeff Bier, BDTI; Mark Birman; David Black; David Boggs; Jim Brady; Forrest Brewer; Aaron Brown, University of California, Berkeley; E. Bugnion, Compaq's Western Research Lab; Alper Buyuktosunoglu, University of Rochester; Mark Callaghan; Jason F. Cantin; Paul Carrick; Chen-Chung Chang; Lei Chen, University of Rochester; Pete Chen; Nhan Chu; Doug Clark, Princeton University; Bob Cmelik; John Crawford; Zarka Cvetanovic; Mike Dahlin, University of Texas, Austin; Merrick Darley; the staff of the DEC Western Research Laboratory; John DeRosa; Lloyd Dickman; J. Ding; Susan Eggers, University of Washington; Wael El-Essawy, University of Rochester; Patty Enriquez, Mills; Milos Ercegovac; Robert Garner; K. Gharachorloo, Compaq's Western Research Lab; Garth Gibson; Ronald Greenberg; Ben Hao; John Henning, Compaq; Mark Hill, University of Wisconsin–Madison; Danny Hillis; David Hodges; Urs Hölzle, Google; David Hough; Ed Hudson; Chris Hughes, University of Illinois em Urbana–Champaign; Mark Johnson; Lewis Jordan; Norm Jouppi; William Kahan; Randy Katz; Ed Kelly; Richard Kessler; Les Kohn; John Kowaleski, Compaq Computer Corp; Dan Lambright; Gary Lauterbach, Sun Microsystems; Corinna Lee; Ruby Lee; Don Lewine; Chao-Huang Lin; Paul Losleben, Defense Advanced Research Projects Agency; Yung-Hsiang Lu; Bob Lucas, Defense Advanced Research Projects Agency; Ken Lutz; Alan Mainwaring, Intel Berkeley Research Labs; Al Marston; Rich Martin, Rutgers; John Mashey; Luke McDowell; Sebastian Mirolo, Trimedia Corporation; Ravi Murthy; Biswadeep Nag; Lisa Noordergraaf, Sun Microsystems; Bob Parker, Defense Advanced Research Projects Agency; Vern Paxson, Center for Internet Research; Lawrence Prince; Steven Przybylski; Mark Pullen, Defense Advanced Research Projects Agency; Chris Rowen; Margaret Rowland; Greg Semeraro, University of Rochester; Bill Shannon; Behrooz Shirazi; Robert Shomler; Jim Slager; Mark Smotherman, Clemson University; o SMT research group, University of Washington; Steve Squires, Defense Advanced Research Projects Agency; Ajay Sreekanth; Darren Staples; Charles Stapper; Jorge Stolfi; Peter Stoll; os estudantes de Stanford e de Berkeley, que deram suporte às nossas primeiras tentativas de escrever este livro; Bob Supnik; Steve Swanson; Paul Taysom; Shreekant Thakkar; Alexander Thomasian, New Jersey Institute of Technology; John Toole, Defense Advanced Research Projects Agency; Kees A. Vissers, Trimedia Corporation; Willa Walker; David Weaver; Ric Wheeler, EMC; Maurice Wilkes; Richard Zimmerman.

John Hennessy, David Patterson

Introdução

Por Luiz André Barroso, Google Inc.

A primeira edição de *Arquitetura de Computadores: Uma Abordagem Quantitativa*, de Hennessy e Patterson, foi lançada durante meu primeiro ano na universidade. Eu pertenço, portanto, àquela primeira leva de profissionais que aprenderam a disciplina usando este livro como guia. Sendo a perspectiva um ingrediente fundamental para um prefácio útil, eu me encontro em desvantagem, dado o quanto dos meus próprios pontos de vista foram coloridos pelas quatro edições anteriores deste livro. Outro obstáculo para uma perspectiva clara é que a reverência de estudante a esses dois superastros da Ciência da Computação ainda não me abandonou, apesar de (ou talvez por causa de) eu ter tido a chance de conhecê-los nos anos seguintes. Essas desvantagens são mitigadas pelo fato de eu ter exercido essa profissão continuamente desde a primeira edição deste livro, o que me deu a chance de desfrutar sua evolução e relevância duradora.

A última edição veio apenas dois anos depois que a feroz corrida industrial por maior frequência de clock de CPU chegou oficialmente ao fim, com a Intel cancelando o desenvolvimento de seus núcleos únicos de 4 GHz e abraçando as CPUs multicore. Dois anos foi tempo suficiente para John e Dave apresentarem essa história não como uma atualização aleatória da linha de produto, mas como um ponto de inflexão definidor da tecnologia da computação na última década. Aquela quarta edição teve ênfase reduzida no paralelismo em nível de instrução (Instruction-Level Parallelism – ILP) em favor de um material adicional sobre paralelismo, algo em que a edição atual vai além, dedicando dois capítulos ao paralelismo em nível de thread e dados, enquanto limita a discussão sobre ILP a um único capítulo. Os leitores que estão sendo apresentados aos novos engines de processamento gráfico vão se beneficiar especialmente do novo Capítulo 4, que se concentra no paralelismo de dados, explicando as soluções diferentes mas lentamente convergentes oferecidas pelas extensões multimídia em processadores de uso geral e unidades de processamento gráfico cada vez mais programáveis. De notável relevância prática: se você já lutou com a terminologia CUDA, veja a Figura 4.24 (teaser: a memória compartilhada, na verdade, é local, e a memória global se parece mais com o que você consideraria memória compartilhada).

Embora ainda estejamos no meio dessa mudança para a tecnologia multicore, esta edição abarca o que parece ser a próxima grande mudança: computação em nuvem. Nesse caso, a ubiquidade da conectividade à Internet e a evolução de serviços Web atraentes estão trazendo para o centro do palco dispositivos muito pequenos (smartphones, tablets) e muito grandes (sistemas de computação em escala de depósito). O ARM Cortex A8, uma CPU popular para smartphones, aparece na seção "Juntando tudo" do Capítulo 3, e um Capítulo 6 totalmente novo é dedicado ao paralelismo em nível de requisição e dados no contexto dos sistemas de computação em escala de depósito. Neste novo capítulo, John e Dave apresentam esses novos grandes clusters como uma nova classe distinta de computadores – um convite aberto para os arquitetos de computadores ajudarem a moldar

esse campo emergente. Os leitores vão apreciar o modo como essa área evoluiu na última década, comparando a arquitetura do cluster Google descrita na terceira edição com a encanação mais moderna apresentada no Capítulo 6 desta versão.

Aqueles que estão retomando este livro vão poder apreciar novamente o trabalho de dois destacados cientistas da computação que, ao longo de suas carreiras, aperfeiçoaram a arte de combinar o tratamento das ideias com princípios acadêmicos com uma profunda compreensão dos produtos e tecnologias de ponta dessa indústria. O sucesso dos autores nas interações com a indústria não será uma surpresa para aqueles que testemunharam como Dave conduz seus retiros bianuais de projeto, foruns meticulosamente elaborados para extrair o máximo das colaborações acadêmico-industriais. Aqueles que se lembram do sucesso do empreendimento de John com o MIPS ou esbarraram com ele em um corredor no Google (o que às vezes acontece comigo) também não vão se surpreender.

E talvez o mais importante: leitores novos e antigos vão obter aquilo por que pagaram. O que fez deste livro um clássico duradouro foi o fato de que cada edição não é uma atualização, mas uma extensa revisão que apresenta as informações mais atuais e insights incomparáveis sobre esse campo fascinante e rapidamente mutável. Para mim, depois de vinte anos nessa profissão, ele é também outra oportunidade de experimentar aquela admiração de estudante por dois professores notáveis.

Prefácio

Por que escrevemos este livro

Ao longo das cinco edições deste livro, nosso objetivo tem sido descrever os princípios básicos por detrás dos desenvolvimentos tecnológicos futuros. Nosso entusiasmo com relação às oportunidades em arquitetura de computadores não diminuiu, e repetimos o que dissemos sobre essa área na primeira edição: "Essa não é uma ciência melancólica de máquinas de papel que nunca funcionarão. Não! É uma disciplina de interesse intelectual incisivo, que exige o equilíbrio entre as forças do mercado e o custo-desempenho-potência, levando a gloriosos fracassos e a alguns notáveis sucessos".

O principal objetivo da escrita de nosso primeiro livro era mudar o modo como as pessoas aprendiam e pensavam a respeito da arquitetura de computadores. Acreditamos que esse objetivo ainda é válido e importante. Esse campo está mudando diariamente e precisa ser estudado com exemplos e medidas reais sobre computadores reais, e não simplesmente como uma coleção de definições e projetos que nunca precisarão ser compreendidos. Damos boas-vindas entusiasmadas a todos os que nos acompanharam no passado e também àqueles que estão se juntando a nós agora. De qualquer forma, prometemos o mesmo enfoque quantitativo e a mesma análise de sistemas reais.

Assim como nas versões anteriores, nos esforçamos para elaborar uma nova edição que continuasse a ser relevante tanto para os engenheiros e arquitetos profissionais quanto para aqueles envolvidos em cursos avançados de arquitetura e projetos de computador. Assim como os livros anteriores, esta edição visa desmistificar a arquitetura de computadores com ênfase nas escolhas de custo-benefício-potência e bom projeto de engenharia. Acreditamos que o campo tenha continuado a amadurecer, seguindo para o alicerce quantitativo rigoroso das disciplinas científicas e de engenharia bem estabelecidas.

Esta edição

Declaramos que a quarta edição de *Arquitetura de Computadores: Uma Abordagem Quantitativa* podia ser a mais significativa desde a primeira edição, devido à mudança para chips multicore. O feedback que recebemos dessa vez foi de que o livro havia perdido o foco agudo da primeira edição, cobrindo tudo igualmente, mas sem ênfase nem contexto. Estamos bastante certos de que não se dirá isso da quinta edição.

Nós acreditamos que a maior parte da agitação está nos extremos do tamanho da computação, com os dispositivos pessoais móveis (Personal Mobile Devices – PMDs), como telefones celulares e tablets, como clientes e computadores em escala de depósito oferecendo computação na nuvem como servidores. (Bons observadores devem ter notado a dica sobre computação em nuvem na capa do livro.) Estamos impressionados com o tema comum desses dois extremos em custo, desempenho e eficiência energética, apesar de sua diferença em tamanho. Como resultado, o contexto contínuo em cada capítulo é

a computação para PMDs e para computadores em escala de depósito, e o Capítulo 6 é totalmente novo com relação a esse tópico.

O outro tema é o paralelismo em todas as suas formas. Primeiro identificamos os dois tipos de paralelismo em nível de aplicação no Capítulo 1, o *paralelismo em nível de dados* (Data-Level Parallelism – DLP), que surge por existirem muitos itens de dados que podem ser operados ao mesmo tempo, e o *paralelismo em nível de tarefa* (Task-Level Parallelism – TLP), que surge porque são criadas tarefas que podem operar independentemente e, em grande parte, em paralelo. Então, explicamos os quatro estilos arquitetônicos que exploram DLP e TLP: *paralelismo em nível de instrução* (Instruction-Level Parallelism – ILP) no Capítulo 3; *arquiteturas de vetor* e *unidades de processamento gráfico (GPUs)* no Capítulo 4, que foi escrito para esta edição; *paralelismo em nível de thread* no Capítulo 5; e *paralelismo em nível de requisição* (Request-Level Parallelism – RLP), através de computadores em escala de depósito no Capítulo 6, que também foi escrito para esta edição. Nós deslocamos a hierarquia de memória mais para o início do livro (Capítulo 2) e realocamos o capítulo sobre sistemas de armazenamento no Apêndice D. Estamos particularmente orgulhosos do Capítulo 4, que contém a mais clara e mais detalhada explicação já dada sobre GPUs, e do Capítulo 6, que é a primeira publicação dos detalhes mais recentes de um computador em escala de depósito do Google.

Como nas edições anteriores, os primeiros três apêndices do livro fornecem o conteúdo básico sobre o conjunto de instruções MIPS, hierarquia de memória e pipelining aos leitores que não leram livros como *Computer Organization and Design*. Para manter os custos baixos e ainda assim fornecer material suplementar que seja do interesse de alguns leitores, disponibilizamos mais nove apêndices onlines em inglês na página www.elsevier.com.br/hennessy. Há mais páginas nesses apêndices do que neste livro!

Esta edição dá continuidade à tradição de usar exemplos reais para demonstrar as ideias, e as seções "Juntando tudo" são novas – as desta edição incluem as organizações de pipeline e hierarquia de memória do processador ARM Cortex A8, o processador Intel Core i7, as GPUs NVIDIA GTX-280 e GTX-480, além de um dos computadores em escala de depósito do Google.

Seleção e organização de tópicos

Como nas edições anteriores, usamos uma técnica conservadora para selecionar os tópicos, pois existem muito mais ideias interessantes em campo do que poderia ser abordado de modo razoável em um tratamento de princípios básicos. Nós nos afastamos de um estudo abrangente de cada arquitetura, com que o leitor poderia se deparar por aí. Nossa apresentação enfoca os principais conceitos que podem ser encontrados em qualquer máquina nova. O critério principal continua sendo o da seleção de ideias que foram examinadas e utilizadas com sucesso suficiente para permitir sua discussão em termos quantitativos.

Nossa intenção sempre foi enfocar o material que não estava disponível em formato equivalente em outras fontes, por isso continuamos a enfatizar o conteúdo avançado sempre que possível. Na realidade, neste livro existem vários sistemas cujas descrições não podem ser encontradas na literatura. (Os leitores interessados estritamente em uma introdução mais básica à arquitetura de computadores deverão ler *Organização e projeto de computadores: a interface hardware/software.*)

Visão geral do conteúdo

Nesta edição o Capítulo 1 foi aumentado: ele inclui fórmulas para energia, potência estática, potência dinâmica, custos de circuito integrado, confiabilidade e disponibilidade. Esperamos que esses tópicos possam ser usados ao longo do livro. Além dos princípios

quantitativos clássicos do projeto de computadores e medição de desempenho, a seção PIAT foi atualizada para usar o novo benchmark SPECPower.

Nossa visão é de que hoje a arquitetura do conjunto de instruções está desempenhando um papel inferior ao de 1990, de modo que passamos esse material para o Apêndice A. Ele ainda usa a arquitetura MIPS64 (para uma rápida revisão, um breve resumo do ISA MIPS pode ser encontrado no verso da contracapa). Para os fãs de ISAs, o Apêndice K aborda 10 arquiteturas RISC, o 80x86, o VAX da DEC e o 360/370 da IBM.

Então, prosseguimos com a hierarquia de memória no Capítulo 2, uma vez que é fácil aplicar os princípios de custo-desempenho-energia a esse material e que a memória é um recurso essencial para os demais capítulos. Como na edição anterior, Apêndice B contém uma revisão introdutória dos princípios de cache, que está disponível caso você precise dela. O Capítulo 2 discute 10 otimizações avançadas dos caches. O capítulo inclui máquinas virtuais, que oferecem vantagens em proteção, gerenciamento de software e gerenciamento de hardware, e tem um papel importante na computação na nuvem. Além de abranger as tecnologias SRAM e DRAM, o capítulo inclui material novo sobre a memória Flash. Os exemplos PIAT são o ARM Cortex A8, que é usado em PMDs, e o Intel Core i7, usado em servidores.

O Capítulo 3 aborda a exploração do paralelismo em nível de instrução nos processadores de alto desempenho, incluindo execução superescalar, previsão de desvio, especulação, escalonamento dinâmico e multithreading. Como já mencionamos, o Apêndice C é uma revisão do pipelining, caso você precise dele. O Capítulo 3 também examina os limites do ILP. Assim como no Capítulo 2, os exemplos PIAT são o ARM Cortex A8 e o Intel Core i7. Como a terceira edição continha muito material sobre o Itanium e o VLIW, esse conteúdo foi deslocado para o Apêndice H, indicando nossa opinião de que essa arquitetura não sobreviveu às primeiras pretensões.

A crescente importância das aplicações multimídia, como jogos e processamento de vídeo, também aumentou a relevância das arquiteturas que podem explorar o paralelismo em nível de dados. Há um crescente interesse na computação usando unidades de processamento gráfico (Graphical Processing Units – GPUs). Ainda assim, poucos arquitetos entendem como as GPUs realmente funcionam. Decidimos escrever um novo capítulo em grande parte para desvendar esse novo estilo de arquitetura de computadores. O Capítulo 4 começa com uma introdução às arquiteturas de vetor, que serve de base para a construção de explicações sobre extensões de conjunto de instrução SIMD e GPUS (o Apêndice G traz mais detalhes sobre as arquiteturas de vetor). A seção sobre GPUs foi a mais difícil de escrever – foram feitas muitas tentativas para obter uma descrição precisa que fosse também fácil de entender. Um desafio significativo foi a terminologia. Decidimos usar nossos próprios termos e, ao traduzi-los, estabelecer uma relação entre eles e os termos oficiais da NVIDIA (uma cópia dessa tabela pode ser encontrada no verso das capas). Esse capítulo apresenta o modelo roofline de desempenho, usando-o para comparar o Intel Core i7 e as GPUs NVIDIA GTX 280 e GTX 480. O capítulo também descreve a GPU Tegra 2 para PMDs.

O Capítulo 5 descreve os processadores multicore. Ele explora as arquiteturas de memória simétricas e distribuídas, examinando os princípios organizacionais e o desempenho. Os tópicos de sincronismo e modelos de consistência de memória vêm em seguida. O exemplo é o Intel Core i7.

Como já mencionado, o Capítulo 6 descreve o mais novo tópico em arquitetura de computadores: os computadores em escala de depósito (Warehouse-Scale Computers – WSCs). Com base na ajuda de engenheiros da Amazon Web Services e Google, esse capítulo integra

detalhes sobre projeto, custo e desempenho dos WSCs que poucos arquitetos conhecem. Ele começa com o popular modelo de programação MapReduce antes de descrever a arquitetura e implementação física dos WSCs, incluindo o custo. Os custos nos permitem explicar a emergência da computação em nuvem, porque pode ser mais barato usar WSCs na nuvem do que em seu datacenter local. O exemplo PIAT é uma descrição de um WSC Google que inclui informações publicadas pela primeira vez neste livro.

Isso nos leva aos Apêndices A a L. O Apêndice A aborda os princípios de ISAs, incluindo MIPS64, e o Apêndice K descreve as versões de 64 bits do Alpha, MIPS, PowerPC e SPARC, além de suas extensões de multimídia. Ele inclui também algumas arquiteturas clássicas (80x86, VAX e IBM 360/370) e conjuntos de instruções embutidas populares (ARM, Thumb, SuperH, MIPS16 e Mitsubishi M32R). O Apêndice H está relacionado a esses conteúdos, pois aborda arquiteturas e compiladores para ISAs VLIW.

Como já dissemos, os Apêndices B e C são tutoriais sobre conceitos básicos de pipelining e caching. Os leitores relativamente iniciantes em caching deverão ler o Apêndice B antes do Capítulo 2, e os novos em pipelining deverão ler o Apêndice C antes do Capítulo 3.

O Apêndice D, "Sistemas de Armazenamento", traz uma discussão maior sobre confiabilidade e disponibilidade, um tutorial sobre RAID com uma descrição dos esquemas RAID 6, e estatísticas de falha de sistemas reais raramente encontradas. Ele continua a fornecer uma introdução à teoria das filas e benchmarks de desempenho de E/S. Nós avaliamos o custo, o desempenho e a confiabilidade de um cluster real: o Internet Archive. O exemplo "Juntando tudo" é o arquivador NetApp FAS6000.

O Apêndice E, elaborado por Thomas M. Conte, consolida o material embutido em um só lugar.

O Apêndice F, sobre redes de interconexão, foi revisado por Timothy M. Pinkston e José Duato. O Apêndice G, escrito originalmente por Krste Asanovic, inclui uma descrição dos processadores vetoriais. Esses dois apêndices são parte do melhor material que conhecemos sobre cada tópico.

O Apêndice H descreve VLIW e EPIC, a arquitetura do Itanium.

O Apêndice I descreve as aplicações de processamento paralelo e protocolos de coerência para o multiprocessamento de memória compartilhada em grande escala. O Apêndice J, de David Goldberg, descreve a aritmética de computador.

O Apêndice L agrupa as "Perspectivas históricas e referências" de cada capítulo em um único apêndice. Ele tenta dar o crédito apropriado às ideias presentes em cada capítulo e o contexto histórico de cada invenção. Gostamos de pensar nisso como a apresentação do drama humano do projeto de computador. Ele também dá referências que o aluno de arquitetura pode querer pesquisar. Se você tiver tempo, recomendamos a leitura de alguns dos trabalhos clássicos dessa área, que são mencionados nessas seções. É agradável e educativo ouvir as ideias diretamente de seus criadores. "Perspectivas históricas" foi uma das seções mais populares das edições anteriores.

Navegando pelo texto

Não existe uma ordem melhor para estudar os capítulos e os apêndices, mas todos os leitores deverão começar pelo Capítulo 1. Se você não quiser ler tudo, aqui estão algumas sequências sugeridas:

- *Hierarquia de memória*: Apêndice B, Capítulo 2 e Apêndice D
- *Paralelismo em nível de instrução*: Apêndice C, Capítulo 3, e Apêndice H
- *Paralelismo em nível de dados*: Capítulos 4 e 6, Apêndice G
- *Paralelismo em nível de* thread: Capítulo 5, Apêndices F e I

- *Paralelismo em nível de requisição*: Capítulo 6
- *ISA*: Apêndices A e K

O Apêndice E pode ser lido a qualquer momento, mas pode ser mais bem aproveitado se for lido após as sequências de ISA e cache. O Apêndice J pode ser lido sempre que a aritmética atraí-lo. Você deve ler a parte correspondente ao Apêndice L depois de finalizar cada capítulo.

Estrutura dos capítulos

O material que selecionamos foi organizado em uma estrutura coerente, seguida em todos os capítulos. Começamos explorando as ideias de um capítulo. Essas ideias são seguidas pela seção "Questões cruzadas", que mostra como as ideias abordadas em um capítulo interagem com as dadas em outros capítulos. Isso é seguido pela "Juntando tudo", que une essas ideias, mostrando como elas são usadas em uma máquina real.

Na sequência vem a seção "Falácias e armadilhas", que permite aos leitores aprender com os erros de outros. Mostramos exemplos de enganos comuns e armadilhas arquitetônicas que são difíceis de evitar, mesmo quando você sabe que estão à sua espera. "Falácias e armadilhas" é uma das seções mais populares do livro. Cada capítulo termina com uma seção de "Comentários finais".

Estudos de caso com exercícios

Cada capítulo termina com estudos de caso e exercícios que os acompanham. Criados por especialistas do setor e acadêmicos, os estudos de caso exploram os principais conceitos do capítulo e verificam o conhecimento dos leitores por meio de exercícios cada vez mais desafiadores. Provavelmente, os instrutores vão achar os estudos de caso detalhados e robustos o bastante para permitir que os leitores criem seus próprios exercícios adicionais.

A numeração de cada exercício (<capítulo.seção >) indica a seção de maior relevância para completá-lo. Esperamos que isso ajude os leitores a evitarem exercícios relacionados a alguma seção que ainda não tenham lido, além de fornecer a eles um trecho para revisão. Os exercícios possuem uma classificação para dar aos leitores uma ideia do tempo necessário para concluí-los:

[10] Menos de 5 minutos (para ler e entender)
[15] 5-15 minutos para dar uma resposta completa
[20] 15-20 minutos para dar uma resposta completa
[25] 1 hora para dar uma resposta completa por escrito
[30] Pequeno projeto de programação: menos de 1 dia inteiro de programação
[40] Projeto de programação significativo: 2 semanas
[Discussão] Tópico para discussão com outros

As soluções para estudos de caso e exercícios estarão disponíveis em inglês para os instrutores que se registrarem na página do livro (www.elsevier.com.br/hennessy)

Material complementar

Uma variedade de recursos está disponível online em www.elsevier.com.br/hennessy, incluindo:

- apêndices de referência – alguns com autoria de especialistas sobre o assunto, convidados – abordando diversos tópicos avançados;
- material de perspectivas históricas que explora o desenvolvimento das principais ideias apresentadas em cada um dos capítulos do texto;

- slides para o instrutor em PowerPoint;
- figuras do livro nos formatos PDF, EPS e PPT;
- links para material relacionado na Web;
- lista de erratas.

Novos materiais e links para outros recursos disponíveis na Web serão adicionados regularmente.

Ajudando a melhorar este livro

Finalmente, é possível ganhar dinheiro lendo este livro (Isso é que é custo-desempenho!). Se você ler os "Agradecimentos", a seguir, verá que nos esforçamos muito para corrigir os erros. Como um livro passa por muitas reimpressões, temos a oportunidade de fazer várias correções. Por isso, se você descobrir qualquer bug extra, entre em contato com a editora norte-americana pelo e-mail <ca5comments@mkp.com>.

Comentários finais

Mais uma vez, este livro é resultado de uma verdadeira coautoria: cada um de nós escreveu metade dos capítulos e uma parte igual dos apêndices. Não podemos imaginar quanto tempo teria sido gasto sem alguém fazendo metade do trabalho, servindo de inspiração quando a tarefa parecia sem solução, proporcionando um insight-chave para explicar um conceito difícil, fazendo críticas aos capítulos nos fins de semana e se compadecendo quando o peso de nossas outras obrigações tornava difícil continuar escrevendo (essas obrigações aumentaram exponencialmente com o número de edições, como mostra o minicurriculum de cada um). Assim, mais uma vez, compartilhamos igualmente a responsabilidade pelo que você está para ler.

John Hennessy & David Patterson

Sumário