Distributed and Cloud Computing
Distributed and Cloud Computing
From Parallel Processing to the Internet of Things

Kai Hwang
Geoffrey C. Fox
Jack J. Dongarra
This book is dedicated to our wives: Jennifer, Judy, and Sue; and to our children: Tony, Andrew, Katherine, Annie, Alexis, James, Heather, Pamela, Oliver, Nick, Ben, and Katie.

KH, GF, and JD
PART 1 SYSTEMS MODELING, CLUSTERING, AND VIRTUALIZATION 1

CHAPTER 1 Distributed System Models and Enabling Technologies .......................... 3
Summary ......................................................................................................................... 4

1.1 Scalable Computing over the Internet ................................................................. 4
1.1.1 The Age of Internet Computing .................................................................... 4
1.1.2 Scalable Computing Trends and New Paradigms ........................................ 8
1.1.3 The Internet of Things and Cyber-Physical Systems ................................... 11

1.2 Technologies for Network-Based Systems ....................................................... 13
1.2.1 Multicore CPUs and Multithreading Technologies .................................... 14
1.2.2 GPU Computing to Exascale and Beyond .................................................. 17
1.2.3 Memory, Storage, and Wide-Area Networking ........................................... 20
1.2.4 Virtual Machines and Virtualization Middleware ...................................... 22
1.2.5 Data Center Virtualization for Cloud Computing ....................................... 25

1.3 System Models for Distributed and Cloud Computing .................................... 27
1.3.1 Clusters of Cooperative Computers ............................................................ 28
1.3.2 Grid Computing Infrastructures .................................................................. 29
1.3.3 Peer-to-Peer Network Families .................................................................... 32
1.3.4 Cloud Computing over the Internet ............................................................. 34

1.4 Software Environments for Distributed Systems and Clouds ........................... 36
1.4.1 Service-Oriented Architecture (SOA) ......................................................... 37
1.4.2 Trends toward Distributed Operating Systems ........................................... 40
1.4.3 Parallel and Distributed Programming Models ........................................... 42

1.5 Performance, Security, and Energy Efficiency ............................................... 44
1.5.1 Performance Metrics and Scalability Analysis ............................................ 45
1.5.2 Fault Tolerance and System Availability .................................................... 48
1.5.3 Network Threats and Data Integrity ............................................................ 49
1.5.4 Energy Efficiency in Distributed Computing ............................................ 51

1.6 Bibliographic Notes and Homework Problems ............................................... 55
Acknowledgments ......................................................................................................... 56
References ..................................................................................................................... 56
Homework Problems ..................................................................................................... 58
CHAPTER 2 Computer Clusters for Scalable Parallel Computing

Summary ..................................................................................................................... 66

2.1 Clustering for Massive Parallelism ........................................................................ 66
   2.1.1 Cluster Development Trends ........................................................................ 66
   2.1.2 Design Objectives of Computer Clusters .................................................... 68
   2.1.3 Fundamental Cluster Design Issues ............................................................ 69
   2.1.4 Analysis of the Top 500 Supercomputers ................................................... 71

2.2 Computer Clusters and MPP Architectures ........................................................... 75
   2.2.1 Cluster Organization and Resource Sharing ................................................. 76
   2.2.2 Node Architectures and MPP Packaging ..................................................... 77
   2.2.3 Cluster System Interconnects ....................................................................... 80
   2.2.4 Hardware, Software, and Middleware Support ........................................... 83
   2.2.5 GPU Clusters for Massive Parallelism ....................................................... 83

2.3 Design Principles of Computer Clusters ................................................................ 87
   2.3.1 Single-System Image Features .................................................................... 87
   2.3.2 High Availability through Redundancy ....................................................... 95
   2.3.3 Fault-Tolerant Cluster Configurations ....................................................... 99
   2.3.4 Checkpointing and Recovery Techniques ................................................. 101

2.4 Cluster Job and Resource Management .................................................................. 104
   2.4.1 Cluster Job Scheduling Methods ................................................................. 104
   2.4.2 Cluster Job Management Systems ............................................................... 107
   2.4.3 Load Sharing Facility (LSF) for Cluster Computing ..................................... 109
   2.4.4 MOSIX: An OS for Linux Clusters and Clouds ......................................... 110

2.5 Case Studies of Top Supercomputer Systems ....................................................... 112
   2.5.1 Tianhe-1A: The World Fastest Supercomputer in 2010 .............................. 112
   2.5.2 Cray XT5 Jaguar: The Top Supercomputer in 2009 ..................................... 116
   2.5.3 IBM Roadrunner: The Top Supercomputer in 2008 ..................................... 119

2.6 Bibliographic Notes and Homework Problems .................................................... 120

Acknowledgments ..................................................................................................... 121

References ................................................................................................................ 121

Homework Problems ............................................................................................... 122

CHAPTER 3 Virtual Machines and Virtualization of Clusters and Data Centers ......... 129

Summary ..................................................................................................................... 130

3.1 Implementation Levels of Virtualization ............................................................... 130
   3.1.1 Levels of Virtualization Implementation .................................................... 130
   3.1.2 VMM Design Requirements and Providers .............................................. 133
   3.1.3 Virtualization Support at the OS Level ..................................................... 135
   3.1.4 Middleware Support for Virtualization ..................................................... 138

3.2 Virtualization Structures/Tools and Mechanisms ............................................... 140
   3.2.1 Hypervisor and Xen Architecture .............................................................. 140
   3.2.2 Binary Translation with Full Virtualization .............................................. 141
   3.2.3 Para-Virtualization with Compiler Support ............................................. 143
PART 2  COMPUTING CLOUDS, SERVICE-ORIENTED ARCHITECTURE, AND PROGRAMMING  189

CHAPTER 4  Cloud Platform Architecture over Virtualized Data Centers  191

Summary  192

4.1  Cloud Computing and Service Models  192
  4.1.1  Public, Private, and Hybrid Clouds  192
  4.1.2  Cloud Ecosystem and Enabling Technologies  196
  4.1.3  Infrastructure-as-a-Service (IaaS)  200
  4.1.4  Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)  203

4.2  Data-Center Design and Interconnection Networks  206
  4.2.1  Warehouse-Scale Data-Center Design  206
  4.2.2  Data-Center Interconnection Networks  208
  4.2.3  Modular Data Center in Shipping Containers  211
  4.2.4  Interconnection of Modular Data Centers  212
  4.2.5  Data-Center Management Issues  213

4.3  Architectural Design of Compute and Storage Clouds  215
  4.3.1  A Generic Cloud Architecture Design  215
  4.3.2  Layered Cloud Architectural Development  218
  4.3.3  Virtualization Support and Disaster Recovery  221
  4.3.4  Architectural Design Challenges  225
5.5 Workflow in Service-Oriented Architectures ........................................ 314
  5.5.1 Basic Workflow Concepts .................................................. 315
  5.5.2 Workflow Standards ....................................................... 316
  5.5.3 Workflow Architecture and Specification ..................................... 317
  5.5.4 Workflow Execution Engine ................................................ 319
  5.5.5 Scripting Workflow System Swift ........................................... 321

5.6 Bibliographic Notes and Homework Problems ..................................... 322
  Acknowledgements .............................................................................. 324
  References .......................................................................................... 324
  Homework Problems ........................................................................... 331

CHAPTER 6 Cloud Programming and Software Environments ......................... 335

6.1 Features of Cloud and Grid Platforms .............................................. 336
  6.1.1 Cloud Capabilities and Platform Features .................................... 336
  6.1.2 Traditional Features Common to Grids and Clouds ............................ 336
  6.1.3 Data Features and Databases ............................................... 340
  6.1.4 Programming and Runtime Support ......................................... 341

6.2 Parallel and Distributed Programming Paradigms ................................ 343
  6.2.1 Parallel Computing and Programming Paradigms .............................. 344
  6.2.2 MapReduce, Twister, and Iterative MapReduce ................................ 345
  6.2.3 Hadoop Library from Apache .............................................. 355
  6.2.4 Dryad and DryadLINQ from Microsoft ........................................ 359
  6.2.5 Sawzall and Pig Latin High-Level Languages .................................. 365
  6.2.6 Mapping Applications to Parallel and Distributed Systems ............... 368

6.3 Programming Support of Google App Engine ..................................... 370
  6.3.1 Programming the Google App Engine ....................................... 370
  6.3.2 Google File System (GFS) ................................................. 373
  6.3.3 BigTable, Google’s NOSQL System ........................................ 376
  6.3.4 Chubby, Google’s Distributed Lock Service .................................. 379

6.4 Programming on Amazon AWS and Microsoft Azure .......................... 379
  6.4.1 Programming on Amazon EC2 .............................................. 380
  6.4.2 Amazon Simple Storage Service (S3) ........................................ 382
  6.4.3 Amazon Elastic Block Store (EBS) and SimpleDB ............................. 383
  6.4.4 Microsoft Azure Programming Support ...................................... 384

6.5 Emerging Cloud Software Environments .......................................... 387
  6.5.1 Open Source Eucalyptus and Nimbus ...................................... 387
  6.5.2 OpenNebula, Sector/Sphere, and OpenStack ................................. 389
  6.5.3 Manjrasoft Aneka Cloud and Appliances ..................................... 393

6.6 Bibliographic Notes and Homework Problems ..................................... 399
  Acknowledgement ................................................................................. 399
  References .......................................................................................... 399
  Homework Problems ........................................................................... 405
CHAPTER 7  Grid Computing Systems and Resource Management

Summary

7.1  Grid Architecture and Service Modeling
   7.1.1  Grid History and Service Families
   7.1.2  CPU Scavenging and Virtual Supercomputers
   7.1.3  Open Grid Services Architecture (OGSA)
   7.1.4  Data-Intensive Grid Service Models

7.2  Grid Projects and Grid Systems Built
   7.2.1  National Grids and International Projects
   7.2.2  NSF TeraGrid in the United States
   7.2.3  DataGrid in the European Union
   7.2.4  The ChinaGrid Design Experiences

7.3  Grid Resource Management and Brokering
   7.3.1  Resource Management and Job Scheduling
   7.3.2  Grid Resource Monitoring with CGSP
   7.3.3  Service Accounting and Economy Model
   7.3.4  Resource Brokering with Gridbus

7.4  Software and Middleware for Grid Computing
   7.4.1  Open Source Grid Middleware Packages
   7.4.2  The Globus Toolkit Architecture (GT4)
   7.4.3  Containers and Resources/Data Management
   7.4.4  The ChinaGrid Support Platform (CGSP)

7.5  Grid Application Trends and Security Measures
   7.5.1  Grid Applications and Technology Fusion
   7.5.2  Grid Workload and Performance Prediction
   7.5.3  Trust Models for Grid Security Enforcement
   7.5.4  Authentication and Authorization Methods
   7.5.5  Grid Security Infrastructure (GSI)

7.6  Bibliographic Notes and Homework Problems

Acknowledgments

References

Homework Problems

CHAPTER 8  Peer-to-Peer Computing and Overlay Networks

Summary

8.1  Peer-to-Peer Computing Systems
   8.1.1  Basic Concepts of P2P Computing Systems
   8.1.2  Fundamental Challenges in P2P Computing
   8.1.3  Taxonomy of P2P Network Systems

8.2  P2P Overlay Networks and Properties
   8.2.1  Unstructured P2P Overlay Networks
9.4.2 Retailing and Supply-Chain Management ........................................ 591
9.4.3 Smart Power Grid and Smart Buildings ............................................ 594
9.4.4 Cyber-Physical System (CPS) ........................................................ 595

9.5 Online Social and Professional Networking ........................................ 597
  9.5.1 Online Social Networking Characteristics ......................................... 597
  9.5.2 Graph-Theoretic Analysis of Social Networks ....................................... 600
  9.5.3 Communities and Applications of Social Networks ............................... 603
  9.5.4 Facebook: The World’s Largest Social Network .................................... 608
  9.5.5 Twitter for Microblogging, News, and Alert Services ............................ 611

9.6 Bibliographic Notes and Homework Problems ........................................ 614
  Acknowledgements .................................................................................. 614
  References ............................................................................................. 614
  Homework Problems ................................................................................ 618

Index ........................................................................................................... 623
ABOUT THE BOOK

Over the past three decades, parallel processing and distributed computing have emerged as a well-developed field in computer science and information technology. Many universities and colleges are now offering standard courses in this field. However, the instructors and students are still in search of a comprehensive textbook that integrates computing theories and information technologies with the design, programming, and application of distributed systems. This book is designed to meet these demands. The book can be also used as a major reference for professionals working in this field.

The book addresses the latest advances in hardware and software, system architecture, new programming paradigms, and ecosystems emphasizing both speed performance and energy efficiency. These latest developments explain how to create high-performance clusters, scalable networks, automated data centers, and high-throughput cloud/grid systems. We also cover programming, and the use of distributed or cloud systems in innovative Internet applications. The book aims to transform traditional multiprocessors and multi-computer clusters into web-scale grids, clouds, and P2P networks for ubiquitous use in the future Internet, including large-scale social networks and the Internet of things that are emerging rapidly in recent years.

A GLANCE AT THE CONTENTS

We have included many milestone developments in a single volume. We present the contributions not only from our own research groups but also from leading research peers in the U.S., China, and Australia. Collectively, this group of authors and contributors summarize the progress that has been made in recent years, ranging from parallel processing to distributed computing and the future Internet.

Starting with an overview of modern distributed models, the text exposes the design principles, system architectures and innovative applications of parallel, distributed, and cloud computing systems. This book attempts to integrate parallel processing technologies with the network-based distributed systems. The book emphasizes scalable physical systems and virtualized data centers and cloud systems for research, e-commerce, social networking, supercomputing, and more applications, using concrete examples from open-source and commercial vendors.

The nine chapters are divided into three Parts: Part 1 covers system models and enabling technologies, including clustering and virtualization. Part 2 presents data center design, cloud computing platforms, service-oriented architectures, and distributed programming paradigms and software support. In Part 3, we study computational/data grids, peer-to-peer networks, ubiquitous clouds, the Internet of Things, and social networks.

Cloud computing material is addressed in six chapters (1, 3, 4, 5, 6, 9). Cloud systems presented include the public clouds: Google AppEngine, Amazon Web Service, Facebook, SalesForce.com, and many others. These cloud systems play an increasingly important role in upgrading the web services and Internet applications. Computer architects, software engineers, and system designers may want to explore the cloud technology to build the future computers and Internet-based systems.
KEY FEATURES

- Coverage of modern distributed computing technology including computer clusters, virtualization, service-oriented architecture, massively parallel processors, peer-to-peer systems, cloud computing, social networks, and the Internet of Things.
- Major emphases of the book lie in exploiting the ubiquity, agility, efficiency, scalability, availability, and programmability of parallel, distributed, and cloud computing systems.
- Latest developments in Hardware, Networks, and System Architecture:
  - Multi-core CPUs and Many-Core GPUs (Intel, Nvidia, AMD)
  - Virtual Machines and Virtual Clusters (CoD, Violin, Amazon VPC)
  - Top-500 Architectures (Tianbhe-1A, Jaguar, Roadrunner, etc.)
  - Google AppEngine, Amazon AWS, Microsoft Azure, IBM BlueCloud
  - TerraGrid, DataGrid, ChinaGrid, BOINC, Grid5000 and FutureGrid
  - Chord, Napster, BitTorrent, KaZaA, PPlive, JXTA, and .NET
  - RFID, Sensor Networks, GPS, CPS, and the Internet of Things
  - Facebook, Force.Com, Twitter, SGI Cylone, Nebula, and GoGrid
- Recent advances in paradigms, programming, software and ecosystems:
  - MapReduce, Dryad, Hadoop, MPI, Twister, BigTable, DISC, etc
  - Cloud Service and Trust Models (SaaS, IaaS, PaaS, and PowerTrust)
  - Programming Languages and Protocols (Python, SOAP, UDDI, Pig Latin)
  - Virtualization Software (XEN, KVM, VMWare ESX, etc.)
  - Cloud OS and Meshups (Eucalyptus, Nimbus, OpenNebula, vShere/4, etc.)
  - Service-Oriented Architecture (REST, WS, Web 2.0, OGSA, etc.)
  - Distributed Operating Systems (DCE, Amoeba, and MOSIX)
  - Middleware and Software Libraries (LSF, Globus, Hadoop, Aneka)
- Over 100 examples are illustrated with 300 figures, designed to meet the need of students taking a distributed system course. Each chapter includes exercises and further reading.
- Included are case studies from the leading distributed computing vendors: Amazon, Google, Microsoft, IBM, HP, Sun, Silicon Graphics, Rackspace, SalesForce.com, netSuite, Enomaly, and many more.

READERSHIP AND SUGGESTIONS TO INSTRUCTORS/STUDENTS

The readership includes students taking a distributed systems or distributed computing class. Professional system designers and engineers may find this book useful as a reference to the latest distributed system technologies including clusters, grids, clouds, and the Internet of Things. The book gives a balanced coverage of all of these topics, looking into the future of Internet and IT evolutions.

The nine chapters are logically sequenced for use in an one-semester (45-hour lectures) course for seniors and graduate-level students. For use in a tri-semester system, Chapters 1, 2, 3, 4, 6, and 9 are suitable for a 10-week course (30-hour lectures). In addition to solving homework problems, the students are advised to conduct some parallel and distributed programming experiments on available clusters, grids, P2P, and cloud platforms. Sample projects and a solutions manual will be made available to proven instructors from Morgan Kaufmann, Publishers.
INVITED CONTRIBUTIONS
The book was jointly planned, authored, edited, and proofread by all three lead authors in four years (2007–2011). Over this period, we have invited and received partial contributions and technical assistance from the following scientists, researchers, instructors, and doctoral students from 10 top Universities in the U.S., China, and Australia.

Listed below are the invited contributors to this book. The authorship, contributed sections, and editorship of individual chapters are explicitly identified at the end of each chapter, separately. We want to thank their dedicated work and valuable contributions throughout the courses of repeated writing and revision process. The comments by the anonymous reviewers are also useful to improve the final contents.

Albert Zomaya, Nikzad Rivandi, Young-Choon Lee, Ali Boloori, Reza Moraveji, Javid Taheri, and Chen Wang, Sydney University, Australia
Rajkumar Buyya, University of Melbourne, Australia
Yongwei Wu, Weimin Zheng, and Kang Chen, Tsinghua University, China
Zhenyu Li, Ninghui Sun, Zhiwei Xu, and Gaogang Xie, Chinese Academy of Sciences
Zhibin Yu, Xiaofei Liao and Hai Jin, Huazhong University of Science and Technology
Judy Qiu, Shrideep Pallickara, Marlon Pierce, Suresh Marru, Gregor Laszewski, Javier Diaz, Archit Kulshrestha, and Andrew J. Younge, Indiana University
Michael McLennan, George Adams III, and Gerhard Klimeck, Purdue University
Zhongyuan Qin, Kaikun Dong, Vikram Dixit, Xiaosong Lou, Sameer Kulkarni, Ken Wu, Zhou Zhao, and Lizhong Chen, University of Southern California
Renato Figueiredo, University of Florida
Michael Wilde, University of Chicago.

PERMISSIONS AND ACKNOWLEDGEMENTS
The permission to use copyrighted illustrations is openly acknowledged in the respective figure captions. In particular, we would like to thank Bill Dally, John Hopecroft, Mendel Roseblum, Dennis Gannon, Jon Kleinberg, Rajkumar Buyya, Albert Zomaya, Randel Bryant, Kang Chen, and Judy Qiu for their generosity in allowing us to use their presentation slides, original figures, and illustrated examples in this book.

We want to thank Ian Foster who wrote the visionary Foreword to introduce this book to our readers. The sponsorship by Todd Green and the editorial work of Robyn Day from Morgan Kaufmann Publishers and the production work led by Dennis Troutman of diacrTech are greatly appreciated. Without the collective effort of all of the above individuals, this book might be still in preparation. We hope that our readers will enjoy reading this timely-produced book and give us feedback for amending omissions and future improvements.

Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra
About the Authors

Kai Hwang is a Professor of Electrical Engineering and Computer Science, University of Southern California. Presently, he also serves as an Intellectual Venture-endowed Visiting Chair Professor at Tsinghua University in Beijing. He earned his Ph.D. in EECS from University of California at Berkeley in 1972. He has taught at Purdue University for 12 years prior to joining USC in 1985. Hwang has served as the founding Editor-in-Chief of the Journal of Parallel and Distributed Computing for 26 years. He is a world-renowned scholar and educator in computer science and engineering. He has published 8 books and 220 original papers in computer architecture, digital arithmetic, parallel processing, distributed systems, Internet security, and cloud computing.

Four of his published books: Computer Arithmetic (Wiley, 1978), Computer Architecture and Parallel Processing (McGraw-Hill, 1983), Advanced Computer Architecture (McGraw-Hill, 1993), and Scalable Parallel Computing (McGraw-Hill, 1998) have been translated into Spanish, Japanese, Chinese, and Korean from the English editions. By 2011, his published papers and books were cited over 9,000 times. Dr. Hwang has delivered three dozen keynote addresses in major IEEE/ACM conferences. In 1986, the IEEE Computer Society elevated him to an IEEE Fellow. He received the 2004 Outstanding Achievement Award from China Computer Federation. In May 2011, he received the IPDPS Founders’ Award for his pioneering contributions in the field of parallel processing. For details, visit the web page: http://GridSec.usc.edu/hwang.html.

Geoffrey C. Fox is a Distinguished Professor of Informatics, Computing and Physics and Associate Dean of Graduate Studies and Research in the School of Informatics and Computing, Indiana University. He has taught and led many research groups at Caltech and Syracuse University, previously. He received his Ph.D. from Cambridge University, U.K. Fox is well known for his comprehensive work and extensive publications in parallel architecture, distributed programming, grid computing, web services, and Internet applications. His book on Grid Computing (coauthored with F. Berman and Tony Hey) is widely used by the research community. He has produced over 60 Ph.D. students in physics, computer science, and engineering over the years. Contact him via: gcf@indiana.edu.

Jack J. Dongarra is a University Distinguished Professor of Electrical Engineering and Computer Science, University of Tennessee, a Distinguished Research Staff, Oak Ridge National Laboratory, and a Turning Fellow at the University of Manchester. An ACM/IEEE/SIAM/AAAS Fellow, Dongarra pioneered the areas of supercomputer benchmarks, numerical analysis, PDE solvers, and high-performance computing and published extensively in these areas. He leads the Linpack benchmark evaluation of the Top-500 fastest computers over the years. Based on his high contributions in the supercomputing and high-performance areas, he was elected as a Member of the National Academy of Engineering in the U.S. Contact him at dongarra@eecs.utk.edu.
Richard Feynman, in his wonderful autobiography *Surely You’re Joking, Mr. Feynman*, recounts how at Los Alamos in 1944 he was responsible for supervising the human computers who performed the long and tedious calculations required by the Manhattan Project. Using the mechanical calculators that were then the state of the art, the best human computer could achieve only one addition or multiplication every few seconds. Feynman and his team thus developed methods for decomposing problems into smaller tasks that could be performed simultaneously by different people (they passed cards with intermediate results between people operating adders, multipliers, collators, and sorters); for running multiple computations at once in the same computing complex (they used different color cards); for prioritizing a more important computation (they eliminated cards of other colors); and for detecting and recovering efficiently from errors (relevant cards, and their descendants, were removed, and computations restarted).

Seventy years later, computer architects face similar challenges—and have adopted similar solutions. Individual computing devices are far faster, but physical constraints still limit their speed. Thus, today’s computing landscape is characterized by pervasive parallelism. Individual processors incorporate pipelining, parallel instructions, speculative execution, and multithreading. Essentially every computer system, from the humblest desktop to the most powerful supercomputer, incorporates multiple processors. Designers of future exascale supercomputers, to be capable of $10^{18}$ operations per second, tell us that these computers will need to support $10^7$ concurrent operations.

Parallelism is fundamentally about communication and coordination, and those two activities have also been transformed over the past seventy years by dramatic technological change. Light is no faster, at 8 inches or 20 centimeters per nanosecond in fiber, than in Feynman’s time; one can never expect to send a message in less than 50 milliseconds from Los Angeles to Auckland. But the rate at which data can be transmitted has changed dramatically, from a few characters per second in 1910 (early telegraphs) to thousands of characters per second in 1960 (ARPANET) to more than 10 billion characters per second over optical fibers in 2010.

Quasi-ubiquitous high-speed communications not only allows call centers to be relocated to India, it also allows computation to be moved to centralized facilities that achieve massive economies of scale, and permits enormous quantities of data to be collected and organized to support decision making by people worldwide. Thus, government agencies, research laboratories, and companies who need to simulate complex phenomena create and operate enormous supercomputers with hundreds of thousands of processors. Similarly, companies such as Google, Facebook, and Microsoft who need to process large quantities of data operate numerous massive “cloud” data centers that may each occupy tens of thousands of square feet and contain tens or hundreds of thousands of computers. Like Feynman’s Los Alamos team, these computing complexes provide computing as a service for many people, and must juggle many computations performed for different purposes.

Massive parallelism, ultra-fast communication, and massive centralization are all fundamental to human decision making today. The computations that are used to forecast tomorrow’s weather, index the web, recommend movies, suggest social connections, predict the future state of the stock market, or provide any one of a multitude of other desirable information products are typically distributed over thousands of processors and depend on data collected from sometimes millions of
sources worldwide. Indeed, little of the modern world could function as it does without parallel and distributed computing.

In this pervasively parallel and distributed world, an understanding of distributed computing is surely an essential part of any undergraduate education in computer science. (Indeed, I would argue, an understanding of these topics should be an essential part of any undergraduate education. But I leave that argument for another time.) The most complex computer systems today are no longer individual microprocessors, but entire data centers. The most complex computer programs written today are those that manage or run on data-center-scale systems. A student who graduates with a degree in computer science and does not understand how these systems and programs are constructed is profoundly unprepared to engage productively in the modern workforce.

Hwang, Fox, and Dongarra’s text is thus especially timely. In its three sections, it covers progressively the hardware and software architectures that underpin modern massively parallel computer systems; the concepts and technologies that enable cloud and distributed computing; and advanced topics in distributed computing, including grid, peer-to-peer, and the Internet of Things. In each area, the text takes a systems approach, describing not only concepts but also representative technologies and realistic large-scale distributed computing deployments. Computing is as much an engineering discipline as a science, and these descriptions of real systems will both prepare students to use those systems and help them understand how other architects have maneuvered the constraints associated with large-scale distributed system design.

The text also addresses some of the more challenging issues facing computer science researchers today. To name just two, computers have emerged as a major consumer of electricity, accounting for several percent of all electricity used in the US. (In Japan, it is ironic that following the 2011 tsunami, the large supercomputers that may help prepare for future natural disasters must often be turned off to conserve power.) And, the fact that 98% of the roughly 10 billion processors sold each year are for embedded devices, and that these embedded devices are increasingly communication-enabled, introduces the opportunity and challenge of an “Internet of Things” that will be vastly larger, more complex, and more capable than today’s Internet of People.

I hope that the appearance of this book will stimulate more teaching of distributed computing in universities and colleges—and not just as an optional topic, as is too often the case, but as a core element of the undergraduate curriculum. I hope also that others outside universities will take this opportunity to learn about distributed computing, and more broadly about what computing looks like on the cutting edge: sometimes messy; often complex; but above all tremendously exciting.

Ian Foster
Jackson Hole, Wyoming
August, 2011