1-2011

An Exceptionally Useful Exploration

George K. Thiruvathukal
Loyola University Chicago, gkt@cs.luc.edu

Recommended Citation

George K. Thiruvathukal, "An Exceptionally Useful Exploration," Computing in Science and Engineering, vol. 13, no. 1, pp. 5-8, Jan./Feb. 2011, doi:10.1109/MCSE.2011.3

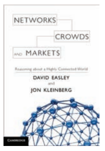
Editors: Larry Engelhardt, lengelhardt@fmarion.edu
Stephen P. Weppner, weppnesp@eckerd.edu

An Exceptionally Useful Exploration of Networks, Crowds, and Markets

A Book that Stands Out in a Crowd

By George K. Thiruvathukal

D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly-Connected World*, Cambridge Univ. Press, 2010, ISBN: 978-0-521-1953-1, 736 pp.



Metcalfe's law (http://en.wikipedia.org/wiki/Metcalfe's_law), which isn't really a law in the scientific sense, tells us that the Internet's usefulness varies by the square of the number of its participants. With this usefulness comes a proportionate increase in complexity. If you're interested in understanding the connected world we live in—a topic not limited to twentysomethings and their Facebook "friend" lists—you'll want to take a look at this exciting new interdisciplinary book by David Easley and Jon Kleinberg.

I found this book when I was asked to teach a course in Science and Society, our interdisciplinary honors course at Loyola University Chicago, which aims to introduce mathematical and scientific ideas to students with greater context than is typical of traditional math and science teaching. This style of learning is more aligned with a classical education, where topics are studied in more integrated ways. When I searched for "networks book" using my "memex" (or memory extender, a.k.a. Google), I came across this book (see www.cs.cornell.edu/home/kleinber/networks-book).

So, for my first-ever offering of a course in our honors program, I decided to focus primarily on networks (the first part of this volume), which has been the subject of many a good book. One need look no further than Nicholas Christakis and James Fowler (*Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives*) and Albert-László Barabási (*Linked: How Everything Is Connected to Everything Else and What It Means*) to find good general books on this subject written in plainspoken language by researchers doing cutting edge work in network analysis and understanding. What these books offer in readability, however, is often lacking in pedagogical soundness, and that's where Easley and Kleinberg come in. In this veritable powerhouse of a textbook, the authors take us through the three areas in the title: networks, crowds, and markets.

The authors draw material from fields such as computer science, mathematics, statistics, economics, sociology, and epidemiology. As they point out in the introduction, the material can be followed linearly, from networking to game theory to their convergence in markets, which are well known for their dynamism—not to mention their obvious interaction with society as a whole. Alternately, the chapters are designed to be relatively self-contained and modular so instructors and students can pick and choose topics of interest (something I've had to do in my course) based on student abilities and interests. As I see it, as a professor or instructor, you can cover as much or as little of this book as you wish, which means you can focus more on ideas and less on quantity.

Graph Theory and Beyond

It's impossible to review every single chapter in a short space, and I think it would miss the point slightly to do so. Beginning with the networking chapters in Part I, I would summarize these as "graph theory for the rest of us." Recalling my graph theory course from graduate school, it goes without saying that there was little context as to why I should care about this highly important subject—a subject that seemingly matters to everything in computer science.

Here, the authors demonstrate numerous examples of what you can model using graph theory: a schism in a karate club, medieval trade routes, international trade, and epidemics (to name just a few). This sets the stage for the rest of the book, where they must introduce and explain a significant number of cross-disciplinary interactions to bring life to the theory.

As an example, the authors introduced how classic social theory and research—the importance of weak ties in forming networks, the notion of homophily (or birds of a feather), and clustering—is crucial to effective network modeling and how the converse is also true. This is graph theory like you've never seen it before: in context!

Judging by the interest in my course—which is largely taken by non-CS and non-math majors—students seem to enjoy seeing their discipline augmented with state-of-the-art mathematical and computer concepts. It might be presumptuous to suggest, but *Networks, Crowds, and Markets* and books like it could become a part of a new kind of computer science. To paraphrase Jeanette Wing's words on "computational thinking," books like this will make computer science the discipline that everyone will want to learn, especially once they know what it can do when combined with other subjects.

The book's next section (Part II) is filled with gems of game theory, a subject that many people refer to colloquially but seldom actually understand. The discussion of Nash equilibrium and strategies/auctions is perhaps one of the clearest I've ever seen and might be a reason to buy this book in its own right. Combining the ideas of graph theory (Part I) and game theory (Part II) leads to markets (Part III), which can be viewed as the beginnings of true modeling and simulation building blocks, which are needed to accomplish the interesting simulations shown in the introductory chapter.

The book's remaining sections can be viewed as "cookbooks" that provide ample material for research or for teaching an advanced course. These chapters all focus on topics of great interest, some of which have made mainstream news. Part IV focuses on the World Wide Web and topics—such as page ranking—that largely aren't well understood by computer science students.

Part V talks about population models, which can be thought of as the theory to support ideas of herds and "crowdsourcing." Part VI deals with more complex structural models, which are needed to understand phenomena such as epidemics (as in the famous susceptible-infected-removed, or SIR, model), while Part VII focuses on aggregate behavior (such as voting systems). Virtually all of these topics could be courses in their own right, and I'm going to feel lucky if I can cover the first two parts of this book in my current course. I'm already thinking that my next course will focus on the other parts of this book.

One thing I found particularly admirable about this book is its open construction, which I think is a glimpse into what textbook publishing can and should become.

A Book Not Prepared in Isolation

One thing I found particularly admirable about this book is its open construction, which I think is a glimpse into what textbook publishing can and should become. The book was actually made available in preprint form for an extended period (and is still available as such). As it turns out, I'm not among the first to use it in a class. Others have used the early drafts and have actually developed lecture notes, including an MIT Open Courseware class that's particularly advanced but nevertheless provides useful notes for instructors.

So, there has been extensive public review of the book by numerous experts in this complex field, which is one of the best ways to ensure quality. I guess it goes without saying that the Internet is an example of the "wisdom in crowds," which the authors surely must have had in mind while working on the manuscript.

That said, one thing I'd like to have seen a bit more of was how to do some of these studies using network modeling software, such as NetLogo (http://ccl.northwestern.edu/netlogo/) and Repast Simphony (http://repast.sourceforge.net), which are two of the major software platforms for doing the kinds of simulations presented in the book's first chapter. Throughout the text, the authors do show examples and clearly mention "agents," referring to the tools required to bring the book's wonderful content to life in practice. To go further and provide an introduction to these tools might be a nice amendment for a future version of the book. Nevertheless, there are plenty of good resources for learning to do agent-based modeling and programming through the NetLogo and Repast project pages.

Given the daunting challenge of covering the book's demanding topics, I'm willing to live with this limitation and view it as a possible blessing in disguise, because I can easily see the book's length doubling if they tried to explain these software packages in any meaningful detail. Readers interested in these topics should visit NetLogo and Repast pages to see how they can turn the outstanding theory in this book into actual code.

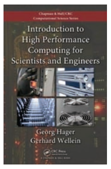
In sum, *Networks, Crowds, and Markets* is an exceptional book. In a time where people are talking about e-books heralding the eventual demise of books in general, this is a book you're going to want to own—in print. The authors seem committed to keeping the prepublication draft available on their site in PDF, which is a great service to humanity and cash-strapped students, so you can even read it for free before purchasing your copy. I managed to learn a great deal about game theory and markets (topics I never studied as a student) and am planning on incorporating these ideas in my teaching and in future research projects.

George K. Thiruvathukal is a computer scientist and interdisciplinary researcher who codirects Loyola University Chicago's new Center for Textual Studies and Digital Humanities. He has written books for Prentice Hall PTR and Sun Microsystems Press and knows how difficult it is to make money writing books, even when you write a good one. Thiruvathukal is associate editor in chief for *CiSE* and for *Computing Now* (an online initiative of the IEEE Computer Society). For more information, please visit http://home.thiruvathukal.com.

A Highly Useful HPC Reference for Scientists and Engineers

By Hang Liu

G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*, Chapman & Hall/CRC Press, Computational Science Series, 2011, ISBN: 978-1-4398-1192-4, 330 pp.



Books on high-performance computing typically fall under two headings. One type is concerned with developing high-performance parallel algorithms for specific tasks, such as linear algebra operations and fast Fourier transforms. The other type discusses how to implement a given algorithm and execute it on different types of modern parallel computer systems. This book by Georg Hager and Gerhard Wellein is a complete and well-organized example of the latter, with a clear main theme: how to achieve and improve scientific computing performance through multiple concurrency and data locality.

Content Overview

In Chapter 1, Hager and Wellein introduce modern processor designs, including chip transistor counts and clock speeds. They start by emphasizing concepts for improving application performance using multiple concurrency and data locality at hardware and instruction level. Examples they discuss include pipelined functional units, superscalar architectures, single instruction multiple data (SIMD), memory cache hierarchies, and multithreaded processors.

Chapters 2 and 3 deal with performance optimization techniques for serial code. In high-performance computing, data access is the most important factor that limits performance. Hence, the focus here is on data access efficiency, and the authors offer many valuable tips. They also introduce balance analysis and bandwidth-based performance modeling, then demonstrate them using the Stream benchmarks to show the hardware's practical capabilities in real applications.

Chapter 4 introduces parallel computing at the application level. It discusses parallelization's benefit and power, as well as the factors that limit parallel performance. These factors include communication/synchronization costs, load balancing, and parallelism's inherent overhead. The authors quantify the details of the parallel performance model, efficiency, speedup, scaling, and scalability metrics in Chapter 5. Readers must understand these essential concepts if they want to

- analyze the parallel performance they can expect,
- interpret the reported parallel performance analysis,
- measure an application's parallel performance, and
- identify opportunities for improving parallel performance in an application.

Having developed all the major concepts of high-performance parallel computing, the authors focus on techniques and tools for developing parallel applications in the rest of the book. For example, Chapters 6 and 9 discuss how to develop parallel applications using the most accepted APIs—that is, OpenMP on shared memory systems and MPI on distributed memory systems—while Chapter 11 focuses on MPI/OpenMP hybrid programming. These three chapters are brief. Although definitely too short for serious developers, the chapters provide a simple yet clear introduction for beginners. This truncation is a wise choice; OpenMP and MPI have been the de facto standards for shared and distributed memory programming for a long time, and rich references for them are available elsewhere at a various sophistication levels.

Highs and Lows

Much of this book is commendable: there are many figures in the chapters and a rich bibliography at the end. The figures make it easy to understand the book's points, especially when presenting performance results. The bibliography is organized by topic, making it easy for readers to check for further references on subjects of interest.

As with all computer science books that involve developing technologies, there will be disappointments when the newest hardware isn't discussed. The authors mention in the preface that they deliberately ignore parallel I/O systems whose efficiency is heavily dependent on hardware specifics. While it's true that parallel I/O is a complex subject, it still deserves an examination. With the increased parallel scale of scientific and engineering computing, I/O is easily a bottleneck to application and system performance. Some relevant I/O performance tips would be a helpful supplement to the book's focus on data access performance. Also, the authors ignore the recent developments of modern accelerator technologies and the partitioned global address space (PGAS) programming models. Currently, many accelerator-enabled clusters are available to research communities. One would hope for a brief outlook section or chapter reviewing the opportunities—and limitations—that these tools offer in terms of performance, programmability, and productivity. Perhaps in the next edition.

Overall, Hager and Wellein have written a great book; it's self-consistent and clear, full of practical and invaluable experiences, and is especially suitable as an in-hand reference for scientists and engineers interested in applying high-performance computing to their research. The case studies and problems at the end of the chapters resonate with practical performance tips and suggestions useful to anyone interested in high-performance computing—including me, a scientist at a supercomputer center who often helps and trains fellow researchers in these techniques.

Hang Liu is a research associate in the Texas Advanced Computing Center at the University of Texas, Austin. His research interests include algorithm development, implementation, validation, and performance optimization to suite large-scale physics calculations on modern high-performance computing architectures. Liu has a PhD in physics from Ohio University. Contact him at hliu@tacc.utexas.edu.