

July 02, 2009

Book Review: Petascale Computing: Algorithms and Applications

by John E. West, for HPCwire

Petascale Computing: Algorithms and Applications, edited by David A. Bader (Chapman & Hall/CRC, 2007), is the first book in CRC's Computational Science Series, edited by Horst Simon at Lawrence Berkeley National Lab. Although the book is a collection of papers, Bader has done an excellent job of creating a compilation that holds together and covers a broad topic very well. At the same time, *Petascale Computing* remains accessible to anyone with HPC or scientific application experience.



While this is a book that just about anyone involved in large-scale technical computing will find valuable, I especially commend the book to center leadership and program managers who, having read it, will find themselves in a better position to ask the questions that matter when planning future hardware and software efforts for their teams.

This book grew out of a February 2006 workshop held at Schloss Dagstuhl in Germany (a remarkably-beautiful "country house" that looks like a palace to me), and consists of 24 standalone chapters that cover a wide variety of application areas and algorithm frontiers. The focus of the book is always on the petascale, with the individual chapters analyzing either specific applications, and what makes them variously well- or ill-suited to large-scale computations, or the algorithms and frameworks that will enable performance on the new machines. Although the book was written before the first system met the petaflops benchmark, we are still

only just at the beginning of the journey through petaflops into exaflops, and the material is very fresh.

Specifically, the book's chapters loosely cover: scalable algorithms for large-scale concurrency (for example, multithreaded graph-theoretic algorithms), specific applications (i.e., in weather and climate, molecular dynamics, biology, ...), tools and programming approaches (Charm++, Chapel, fault-tolerant MPI, and so on), and, throughout, performance analysis.

Given the diversity of the book's material, it would be time-consuming, and pointless given the availability of information on the Internet, to provide a treatment of each of the topics in the individual chapters. For that I refer the reader to his or her favorite bookstore or online resource. Instead let me talk about a few of the chapters that I think provide a good overview

of the value of the book for practitioners at the extremes of technical computing.

The first chapter opens with a detailed examination of the performance of five applications that are candidates for petascale processing (both in terms of their architecture and in terms of the problems they are designed to solve) on five current supercomputing systems. The applications are benchmarked on each system, but the information isn't simply tabulated and presented. Rather the authors dive into each application and talk about the characteristics of each code/hardware combination that drive the measured performance results. This leads in each case to a specific discussion about either hardware or software technologies that will be needed to serve the petascale demands of the scientists that will rely on the results. This chapter also incidentally provides the reader exposure to a range of effective tools and performance evaluation techniques that will inform his own analyses.

As an example, consider these passages from the analysis of the performance characteristics of GTC, a 3D particle-in-cell code for studying magnetic confinement plasmas. After describing how the code decomposes the computational domain, the authors go on to make explicit the hardware implications of the decomposition -- a step that is probably unnecessary for the professional computationalist, but which will nevertheless prove very valuable for the many other HPC professionals thinking about the next generation of hardware and software and planning the investments that will get us there:

Figure 1.1 (a) shows the regular communication structure exhibited by GTC. This particle-in-cell calculation uses a one-dimensional domain decomposition across the toroidal computational grid, causing each processor to exchange data with its two neighbors as particles cross the left and right boundaries.... Therefore, on average each MPI task communicates with 4 neighbors, so much simpler interconnect networks will be adequate for its requirements.

After looking fairly closely at the characteristics of each of the five applications, the authors bring the whole set of results together and draw broad conclusions about these codes and their potential for successful petascale deployment based on the evidence gathered. It's a great chapter to open the book with.

An early example of a chapter that makes a deep dive into a specific application domain is chapter six on the numerical prediction of "high-impact" local weather. This chapter exemplifies one of the real strengths of the book: each domain-specific chapter provides enough context and detail to bring along non-experts in the domain, while still managing to cover enough detail that the reader achieves a working knowledge of the high-level drivers in scientific applications. As a result readers exit these chapters with knowledge of why an application does what it does sufficient to think critically about the impact that specific petascale hardware and software features will have on performance.

In chapter six, for example, the authors briefly introduce operational weather forecasting and describe the current state of the practice, weather forecasting at a resolution of 25 km. As a motivation for increasing this resolution, the authors describe what will be needed to both expand the geographic area covered by predictions and enable prediction in sufficient detail to predict items of high local interest, such as thunderstorms and tornadoes:

Even at 12-km horizontal grid spacing, important weather systems that are directly responsible for meteorological hazards including thunderstorms, heavy precipitation and tornadoes cannot be directly resolved because of their small sizes. For individual storm cells, horizontal grid resolutions of at least 1-km grid are generally believed to be necessary, which even high resolutions are needed to resolve less organized storms and the internal circulation within the storm cells.

The authors then go on to describe the features of these codes that drive hardware

requirements, for example:

An evaluation of the ARPS on scalar and vector-based computers indicates that the nested do-loop structures were able to realize a significant percentage of the peak performance of vector platforms, but on commodity-processor-based platforms the utilization efficiency is typically on 5-15%. The primary difference lies with the memory and memory access speeds. Since weather forecast models are largely memory bound, they contain far more loads/stores than computations and, as currently written, do not reuse in-cache data efficiently.

And this analysis continues for data distribution, load balancing issues, scalability, and so on.

There is also discussion of specific architectures and approaches for reaching the petascale, with emphasis on how these will influence application design and optimization. For example chapter 8 considers reaching beyond the petascale by functionally dividing amenable computations onto separate supercomputers, an approach the authors call "distributed petascale computing." Likewise, chapter 10 talks about the MDGRAPE special-purpose hardware project. Later chapters, for example chapter 21, talk about a specific annotation-based approach for performance portability, an important topic to address if indeed petascale architectures end up being as diverse as many expect.

In chapter 13 the authors frame the discussions about massive concurrency and enormous computer systems in a different way: distributing applications over tens or hundreds of thousands of processors is going to create a driver for applications to recover from hardware faults. This chapter discusses FT-MPI, a fault-tolerant MPI implementation, along with a diskless checkpointing approach that, combined, can provide the application developer a good starting point for developing more robust applications. In any discussion of these technologies, the question of performance overhead will naturally (and properly) arise, and the authors present results in the context of a PCG algorithm.

In fact, throughout this book the focus stays on performance -- not just the numbers, but what drives the numbers to fall out the way they do, and, perhaps just as important, why we need the performance in the first place. The end result educates and informs our journey through petascale and into exascale, while serving to motivate us to travel as fast as we can toward the goal.

[Petascale Computing: Algorithms and Applications \(Chapman & Hall/Crc Computational Science Series\)](#)

Find this article at:

<http://www.hpcwire.com/features/Book-Review-Petascale-Computing-Algorithms-and-Applications-49727607.html?viewAll=y>

Check the box to include the list of links referenced in the article.

Copyright © 1994-2008 Tabor Communications, Inc. All Rights Reserved.