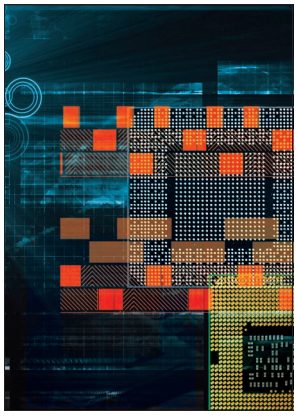
CPUS, GPUS, AND HYBRID COMPUTING



David Brooks
Harvard University

It is my pleasure to introduce this special issue on hybrid CPU/GPU computing. Although general-purpose CPUs have traditionally been the dominant player in both mainstream and high-performance computing systems, recent years have seen a major shift toward GPUs. Originally developed primarily for graphics and video applications, GPUs are increasingly being harnessed using programming languages such as CUDA and OpenCL to provide large performance speedups for general-purpose applications. For example, China's Nebulae supercomputer claims a high spot on the Top 500 supercomputer list while relying on Nvidia's Tesla C2050 GPUs, and the Roadrunner system at Los Alamos National Laboratory relies on IBM's GPU-like Cell processor for much of its computing power. At the other end of the computing spectrum, users of battery-operated mobile devices increasingly desire substantially higher graphics and media performance within tight energy budgets and thermal envelopes. The coming years promise intense development efforts for both CPUs and GPUs, and the next generation of hybrid CPU/GPU chips are now coming to the market with many future systems on the near-term horizon. Coming generations of hybrid systems will include even more specialization in the form of reconfigurable logic (such as field-programmable gate arrays, or FPGAs) or hard-wired hardware accelerators.

Hybrid systems have many advantages that designers hope to capitalize on. GPUs can provide high-capacity number crunching capabilities with a reasonable power and cost budget. The raw flops count for data-parallel workloads running on GPUs is typically much higher than for CPU-based approaches. Energy scaling challenges are critically important to future mobile and high-performance systems, and hybrid systems have a particular advantage here. It is well-known that energy is expended more efficiently by matching the computation to the underlying hardware using specialized architectures. GPUs provide a tangible initial target for this form of hardware specialization, and in the coming years we will see new approaches that enable significant power efficiency benefits by following the paradigm of codesigning the applications and hardware.

Many challenges must be addressed to capitalize on these opportunities. Not all applications will be suited to hybrid systems, and application writers must be cognizant of the underlying architecture to an even greater degree than has been the case for traditional homogenous CPU-based platforms. Providing the mapping from the application layer to the hardware layer is a significant challenge for hybrid systems. In fact, prior heterogeneous systems have provided unparalleled computational horsepower, but it has been notoriously difficult to extract all of this performance in many cases. Advances in programming languages and system software

Published by the IEEE Computer Society

support can address these problems, but many issues must be overcome to make hybrid systems highly productive. Finally, given the enormous costs associated with software development, portability and scalability of large parallel code bases are important challenges that we will face as we move beyond traditional CPU-based system designs.

It was a great honor to serve as guest editor for this *IEEE Micro* special issue. The solicitation sought articles across many areas of interest in the hybrid CPU/GPU space. I was very happy to see a wide range of topics addressed in submissions to this issue. We selected six articles that broadly cover the topic areas of energy and technology scaling for hybrid systems, programming and productivity approaches, and application development for hybrid systems. Although this issue only touches on a few of the challenges and opportunities, I hope that it spurs additional research to address these and other challenges that we face going forward in the space of hybrid systems.

Energy and technology scaling for hybrid systems

In ''GPUs and the Future of Parallel Computing,'' Stephen W. Keckler et al. make the case that energy efficiency, memory bandwidth, and programmability will be the key bottlenecks for highly parallel chips in the future. The article describes Nvidia Research's Echelon project, which targets these three challenges. The project includes the goal of developing a prototype architecture by 2017 capable of peak double-precision throughput of 16 teraflops, a memory bandwidth of 1.6 terabytes per second, and a power budget of under 150 W. This architecture will not be considered an external accelerator, but a combined CPU/GPU platform with a unified memory architecture facilitating programmability and enhancing efficiency.

In ''Multi-GPU DGEMM and High Performance Linpack on Highly Energy-Efficient Clusters,'' David Rohr et al. expand on the theme of GPU energy-efficiency issue at the cluster level. The article discusses multi-GPU support that was added to an implementation of DGEMM (double-precision general matrix-multiply) to allow additional scalability across clusters of GPUs. The article considers the outlook for energy efficiency in multi-GPU clusters and presents several possible directions. This article reinforces the significant opportunities for achieving energy efficiency at the system architecture levels.

Programming and productivity

''PEPPHER: Efficient and Productive Usage of Hybrid Computing Systems'' by Siegfried Benkner et al. describes a large-scale project addressing the problems of programmability and efficiency utilization of hybrid CPU/GPU systems. The system under development provides a software framework to provide portability of parallel codes across systems with different capabilities. Parallel versions of performance-critical codes that target different platforms are developed in multiple ways and deployed across available resources. The article provides an example of how this architecture can be used for Nvidia Tesla and Fermi GPUs. An excellent case study illustrates the concept of the PEPPHER approach.

In ''Implementing Domain-Specific Languages for Heterogeneous Parallel Computing,'' HyoukJoong Lee et al. address the pressing issue of programming language design for heterogeneous systems. The authors discuss the paradigm of domain-specific languages (DSLs), which provide a convenient way for developers to target particular problem domains, usually through APIs that are a close match to a given domain's requirements. DSLs can also target special hardware features of heterogeneous systems by mapping the API to the underlying platform. The article discusses the Delite framework, which lets developers rapidly construct new DSLs to target future hybrid architectures across various problem domains.

Applications

In ''Medical Ultrasound Imaging: To GPU or Not to GPU?'' Hayden K.-H. So et al. explore the computationally challenging and critically important problem of medical ultrasound imaging in GPUs and

hybrid systems. The authors find that GPUs can provide some benefits, but stand-alone GPUs don't entirely solve the problem. Instead, the authors propose a hybrid architecture that combines GPUs with FPGAs, where the FPGA provides substantial preprocessing of the data near the data acquisition source and the GPU provides high-performance computing near the user display.

"Performance Implications of Nonuniform Device Topologies in Scalable Heterogeneous Architectures" by Jeremy S. Meredith et al. explores the implications of heterogeneous GPU systems on parallel scientific applications. Although many high-throughput floating-point codes are well-suited to GPUs, other traditional scientific workloads still require CPU facilities for coarse-grained task scheduling and high-performance I/O. The article describes the authors' experiences with the National Science Foundation's Keeneland system for HPC workloads, highlighting challenges associated with nonuniform data-transfer performance across the node-level topology.

This special issue demonstrates the exciting ongoing research and many future directions in the area of hybrid CPU/GPU systems. It's clear from the diversity of topics in the special issue that future hybrid systems will require significant collaborative efforts from the hardware design, chip and system architecture, system software, and application layers. The articles in this issue highlight some of the challenges within and across each of these layers, and we look forward to seeing progress in all of these areas over the next several years. MICRO

Acknowledgments

I thank Erik Altman for inviting me to guest edit this issue and for providing valuable advice throughout the process. We thank the authors who submitted to this special issue and the many reviewers who provided thoughtful and candid feedback on the articles throughout the review process.

David Brooks is a Gordon McKay Professor of Computer Science in the School of Engineering and Applied Sciences at Harvard University. His research interests include technology-aware computer design, with an emphasis on power-efficient computer architectures and system software for high-performance and embedded systems. Brooks has a PhD in electrical engineering from Princeton University.

Direct questions and comments about this article to David Brooks, Maxwell-Dworkin 141, 33 Oxford St., Cambridge, MA 02138; dbrooks@eecs.harvard.edu.