

## Editorial

# Programming Models, Languages, and Compilers for Manycore and Heterogeneous Architectures

**Sunita Chandrasekaran,<sup>1</sup> Barbara Chapman,<sup>1</sup> Xinmin Tian,<sup>2</sup> and Yonghong Yan<sup>3</sup>**

<sup>1</sup>University of Houston, 4800 Calhoun Road, Houston, TX 77004, USA

<sup>2</sup>Intel Corporation, 2200 Mission College Boulevard, SC12-301, Santa Clara, CA 95052, USA

<sup>3</sup>Oakland University, 2200 N Squirrel Road, Rochester, MI 48309, USA

Correspondence should be addressed to Sunita Chandrasekaran; [sunita@cs.uh.edu](mailto:sunita@cs.uh.edu)

Received 25 March 2015; Accepted 25 March 2015

Copyright © 2015 Sunita Chandrasekaran et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

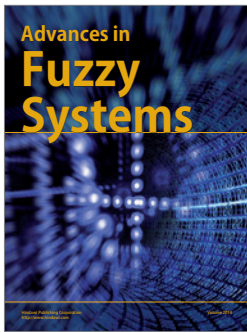
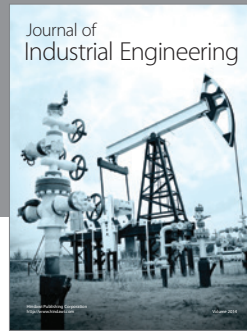
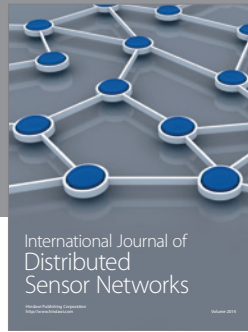
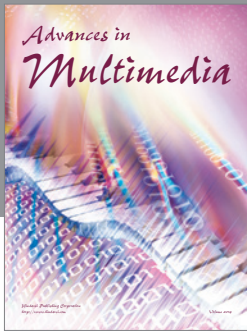
*Aim and Scope.* Hardware is emerging rapidly, core count is increasing, and systems consist of large cluster of nodes. These nodes are becoming more heterogeneous, that is, multicore CPUs attached to accelerators meant to address specific needs of specific computations. The type of parallelism each accelerator offers is quite different from the other. On the other hand, software for such emerging parallel and heterogeneous computing systems is still catching up. The gap between hardware and software is growing; it is a challenge for the software developers to keep up with the hardware advancements. Thus, there is an urgent need to develop and maintain sophisticated software that can not only offer performance productive solutions but also be applicable to a wide range of hardware systems. Some of the promising and widely used programming solutions include directive-based programming models such as OpenMP, vendor-specific programming models such as NVIDIA's CUDA, OpenCL, and recently emerging programming model, OpenACC. This special issue publishes papers on the evaluations of these models for parallel computing with respect to several factors including locality-aware scheduling, data transfer optimizations, SIMD vectorization on Phi coprocessors, and programming multi-GPU.

the manuscripts. To this effort, we would like to thank and acknowledge all external reviewers for spending their valuable time to provide the authors with valuable feedback.

Sunita Chandrasekaran  
Barbara Chapman  
Xinmin Tian  
Yonghong Yan

## Acknowledgments

The editors would like to thank the authors for their tremendous effort and time spent on their respective manuscripts. We relied on peer-review process to uphold the quality of



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

