The Journal of Supercomputing (2019) 75:999–1000
https://doi.org/10.1007/s11227-018-2713-y

Computational and mathematical models meet heterogeneous computing

Diego R. Llanos[1] · **Jesús Vigo-Aguiar**[2]

Published online: 7 December 2018

During the first decade of the twenty-first century, the advent of multicore processing reached its maturity level, with the help of shared-memory programming models such as OpenMP [1], that allows to parallelize both legacy and new C and Fortran applications in a shared-memory environments. Meanwhile, message-passing programming models such as MPI [2] allowed to aggregate multicore systems in larger clusters, which dominated the TOP 500 supercomputing list [3].

However, at that time parallel computing seemed to face some limits that were hard to overcome. Physical limits prevented clock frequencies to increase, and the Law of Diminishing Returns reduced the usefulness of keep adding cores to a multiprocessor. Suddenly, the advent of GPU computing changed the game once again. Being initially developed as a way to accelerate graphical processing, GPUs were reused to speed up certain types of calculations that needed a similar processing to an entire set of independent elements. CUDA [4] programming model allowed programmers to translate to the GPU world many applications, and the TOP 500 list started to show more and more heterogeneous systems that incorporated accelerators to their cluster nodes.

Today, we are facing a new revolution. The popularization of Deep Learning techniques makes different vendors to add specific support for these kinds of algorithms in the accelerators present in the market. Deep Learning is a world that requires high-performance computing, massive parallelism, and it is causing the development of new heterogeneous architectures.

Is in this context of CMMSE conference where the HPC session takes place. Starting in 2000, CMMSE have gathered specialist from all around the globe to discuss recent advances in different topics related to the use of high-performance computing to speed up the execution of mathematical methods. The particular format of the CMMSE conference, with lively discussions in a relaxed atmosphere, allows the presentation of cutting-edge, ongoing work, that receives direct feedback from the audience, thus

✉ Jesús Vigo-Aguiar
jvigo@usal.es

1 Computer Science Department, University of Valladolid, Campus Miguel Delibes s/n., 47011 Valladolid, Spain

2 Facultad de Ciencias, Universidad de Salamanca, 37008 Salamanca, Spain

enriching the research carried out. In an era of big conferences that attract several hundreds of researchers that should compete for the attention of the audience, the CMMSE conference keeps the spirit of specialist meetings, where all the speakers are equally accessible and where interesting discussions are carried out after each presentation.

This special issue of the Journal of Supercomputing contains revised papers, selected from those presented at the mini-symposium on HPC hosted as part of the 17th and 18th Computational and Mathematical Methods in Science and Engineering, held in Cádiz in July 2017 and July 2018. The Associated Editors selected 23% of all the contributions sent to the conference, covering both foundational and practical issues in the use of parallel computing to solve complex applications, for publication in this special issue.

In summary, we believe that the interesting contributions included in this issue are good examples of how the high-performance-computing community works day after day to speed up the performance of different real-world problems, increasing the usefulness of current and future parallel systems and imagining new ways to develop concurrent code.

Acknowledgements We would like to thank all the authors, reviewers, and editors involved in the elaboration of this special issue, including also the reviewers who were involved in the HPC symposium held as part of the CMMSE conference, where preliminary versions of the papers were previously selected. J. Vigo-Aguiar is especially grateful to Professor Hamid Arabnia for his constant encouragement in the development of computational science and his encouragement to find the connection between different fields.

References

1. Dagum L, Menon R (1998) Openmp: an industry standard API for shared-memory programming. IEEE Comput Sci Eng 5(1):46–55
2. Gropp WD, Gropp W, Lusk E, Skjellum A (1999) Using MPI: portable parallel programming with the message-passing interface, vol 1. MIT Press, Cambridge
3. TOP500 Supercomputer sites (2018). www.top500.org. Accessed 5 Dec 2018
4. Sanders J, Kandrot E (2010) CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley Professional, Reading