

Guest Editorial for Programming Models and Algorithms for Data Analysis in HPC Systems

Salvatore Cuomo¹ · Marco Aldinucci² · Massimo Torquati³

Published online: 7 October 2017
© Springer Science+Business Media, LLC 2017

Performance is still the hottest keyword in parallel and distributed systems: performance evaluation, design for performance, performance portability and scalability are just a few of the many possible declinations that nowadays are of paramount scientific importance. To tackle these challenges, system architects, applications programmers and data center managers need methodological tools to fit at best the overall workload and the available architecture, maximizing the overall performances and minimizing overheads, energy consumption or idle time while application developers mainly aim at algorithmic and software oriented performances. Proper methodologies for modeling and analysis are the way to turn complexity into opportunities.

This Special Issue of the International Journal of Parallel Programming welcomes papers that present practical and methodological approaches to analytical and simulative performance evaluation for architecturally complex systems and high-performance parallel and computing algorithm. Successful contributions have been done on specific technologies, applications and innovative solutions to system specifications and algorithmic schemes both.

✉ Salvatore Cuomo
salvatore.cuomo@unina.it

Marco Aldinucci
aldinuc@di.unito.it

Massimo Torquati
torquati@di.unipi.it

¹ Department of Mathematics and Application, University of Naples Federico II, Complesso Universitario di Monte Sant' Angelo, Via Cintia, Naples, Italy

² Department of Computer Science, University of Torino, Turin, Italy

³ Department of Computer Science, University of Pisa, Pisa, Italy

Seven selected papers were accepted for this issue.

The paper [1] presents a system architecture that enhances the working of traditional MapReduce by incorporating parallel processing algorithm. A complete four-tier architecture for efficiently aggregate the data is proposed. This system is evaluated in terms of efficiency by considering the throughput and processing time.

The Overcomplete Local Principal Component Analysis (OLPCA) method for image denoising and its main issues are presented in [2]. A fine-to-coarse parallelization strategy on a parallel hybrid architecture is proposed. Experimental results show improvements in terms of execution time with a promising speed up with respect to the CPU and our old GPU versions.

The impact of package level cohesion metrics on reusability for Aspect Oriented Systems (AOS) is analysed in the paper [3]. A package for cohesion measure, PCohA, is implemented. The proposed package cohesion measure is found to be a useful indicator of external quality factors such as reusability. The proposed metric is also established as a better predictor of code reusability than the existing cohesion measures. The work discussed in this paper can be used for designing high quality software by developing new package level metrics for other quality attributes.

The paper [4] describes a generic and straightforward algorithm, MeshCleaner, for cleaning large Finite Element meshes. The presented mesh cleaning algorithm is composed of (1) the stage of compacting and reordering nodes and (2) the stage of updating mesh topology. The basic ideas for performing the above two stages efficiently both in sequential and in parallel are introduced. Experimental results indicate that the algorithm MeshCleaner is capable of cleaning large meshes very efficiently, both in sequential and in parallel.

In [5] for several computing platforms the efficiency of a pleasingly parallel application is studied. A real-world problem, i.e., Monte-Carlo numerical simulations of stratospheric balloon envelope drift descent is considered. We detail the optimization of the SIMD parallel codes on the K40 and K80 GPUs as well as on the Intel Xeon Phi. The experiments show that GPU and MIC permit to decrease computing time by non negligible factors, which finally allow to use these devices in operational conditions.

The paper [6] deals with the following scheduling problem: an infinite number of tasks must be scheduled for processing on a finite number of heterogeneous machines, such as all tasks are sent to execution with a minimum delay. The proposed model had a starting point in two known bounded number of processors algorithms: Modified Critical Path and Highest Level First With Estimated Times. Regarding the implementation, a simulator was used to analyze and design the scheduling algorithms.

Finally, in [7] an efficient and real-time Big Data stream processing approach mapping Hadoop MapReduce equivalent mechanism on graphics processing units (GPUs) is presented. Parallel and distributed environments of Hadoop ecosystem and a real-time streaming processing tool are integrated. A MapReduce equivalent algorithm for GPUs

for a statistical parameter calculation by dividing overall Big Data files into fixed-size blocks is designed. Results show that the proposed system working with Spark on top and GPUs under the parallel and distributed environment of Hadoop ecosystem is more efficient as compared to existing standalone CPU-based MapReduce implementation.

References

1. Ahmad, A., Paul, A., Din, S., Rathore, M.M., Choi, G.S., Jeon, G.: Multilevel data processing using parallel algorithms for analyzing big data in high-performance computing. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0498-x](https://doi.org/10.1007/s10766-017-0498-x)
2. De Michele, P., Maiorano, F., Marcellino, L., Piccialli, F.: A GPU implementation of OLPCA method in hybrid environment. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0505-2](https://doi.org/10.1007/s10766-017-0505-2)
3. Kaur, P.J., Kaushal, S., Sangaiah, A.K., Piccialli, F.: A framework for assessing reusability using package cohesion measure in aspect oriented systems. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0501-6](https://doi.org/10.1007/s10766-017-0501-6)
4. Mei, G., Cuomo, S., Tian, H., Xu, N., Peng, L.: Meshcleaner: A generic and straightforward algorithm for cleaning finite element meshes. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0507-0](https://doi.org/10.1007/s10766-017-0507-0)
5. Plazolles, B., El Baz, D., Spel, M., Rivola, V., Gegout, P.: SIMD Monte–Carlo numerical simulations accelerated on GPU and xeon phi. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0509-y](https://doi.org/10.1007/s10766-017-0509-y)
6. Popa, E., Iacono, M., Pop, F.: Adapting MCP and HLFET algorithms to multiple simultaneous scheduling. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0516-z](https://doi.org/10.1007/s10766-017-0516-z)
7. Rathore, M.M., Son, H., Ahmad, A., Paul, A., Jeon, G.: Real-time big data stream processing using GPU with spark over hadoop ecosystem. *Int. J. Parallel Program.* (2017). doi:[10.1007/s10766-017-0513-2](https://doi.org/10.1007/s10766-017-0513-2)