Received: 25 February 2019 | Accepted: 7 March 2019
DOI: 10.1002/cpe.5260

WILEY

EDITORIAL

Algorithmic advances in parallel architectures and energy-efficient computing

This special issue of *Concurrency and Computation: Practice and Experience* contains revised and extended versions of selected papers presented at the 12th International Conference on Parallel Processing and Applied Mathematics (PPAM 2017), which was held on September 10-13, 2017, in Lublin, Poland.

PPAM is a biennial series of international conferences dedicated to exchanging ideas between researchers involved in parallel and distributed computing, including theory and applications, as well as applied and computational mathematics. The focus of PPAM 2017 was on models, algorithms, and software tools that facilitate efficient and convenient use of modern parallel and distributed computing systems, as well as on large-scale applications, including data-intensive and machine learning problems.

PPAM 2017 was organized by the Department of Computer and Information Science of Czestochowa University of Technology in Czestochowa, Poland, together with Maria Curie-Sklodowska University in Lublin, Poland, under the patronage of the Committee of Informatics of the Polish Academy of Sciences, in cooperation with the ICT COST Action IC1305 "Network for Sustainable Ultrascale Computing (NESUS)".

This meeting gathered more than 170 participants from 25 countries. The accepted papers were presented at the regular tracks of the PPAM 2017 conference, as well as during the workshops. A strict reviewing process, with each submission evaluated by at least three reviewers, resulted in acceptance of 100 contributed papers for publication in the conference proceedings, while approximately 42% of the submissions were rejected. For regular tracks, 49 papers were selected from 98 submissions, resulting in an acceptance rate of 50%.

Based on the results of the reviews, selected papers were recommended for a special journal issue. Besides quality, another important criterion for selection was each paper contribution to thematic consistency of the issue. The main focus of this special issue is on algorithmic advances in matching the software properties to the targeted parallel architecture, including graphics processing unit (GPU) accelerators and clusters. These advances are crucial for parallelizing successfully such complex applications as simulating granular flows, solving nonsingular systems, electronic transport simulations, solving three-dimensional fractional power diffusion problems, dynamic programming, computer graphics, parallel event-driven simulation, and others. A complementary topic of this issue is energy-efficient computing, since the energy consumption has become a limiting factor for high-performance computing (HPC) applications in recent years. The authors of selected papers were contacted after the conference and invited to submit revised and extended versions of their works. These new versions were reviewed independently again by at least three reviewers. Finally, ten contributions were accepted for publication. They are summarized below.

The work of Krestenitis and Weinzierl[1] focuses on simulating granular flows using discrete element method models. This problem is computationally challenging— a bottleneck arises when identifying all particle contact points per time steps. To introduce concurrency to particle comparisons, while keeping their number low, the authors propose a tree-based multilevel metadata structure to manage the particles, as well as a novel scheme of identifying the contact points. Furthermore, a novel adaptivity criterion allows an explicit time stepping technique to work with comparably large time steps. The fusion of the proposed developments yields promising speedups for maximally asynchronous task-based realizations. This work shows that new computer architectures can push the boundary of such many-particle simulations by choosing the right data structures and data processing schemes.

An efficient algorithm for the parallel robust solution of triangular linear systems is presented in the paper by Mikkelsen et al[2]. Such systems are central to the solution of general linear systems and computation of eigenvectors, using either forward or backward substitution. However, there are well-conditioned systems for which substitution fails due to overflow. This paper presents novel algorithms that are blocked and parallel, while dynamically scaling the solution and right-hand side values to avoid overflows. A new task-based parallel robust solver Kiya is developed and compared against LAPACK solvers. When there are many complex right-hand sides, Kiya performs significantly better than the robust solver DLATRS and is not significantly slower than the nonrobust solver DTRSM.

The algorithm developed in the work of Spellacy et al[3] extends previous work on inversion of block tridiagonal matrices from the Hermitian/symmetric case to the general case, with variable sub-block sizes. The presented investigation is motivated by the requirements of atomic and molecular–scale electronic transport simulations, in particular, the SMEAGOL electronic transport code. A parallel divide-and-conquer approach is used to develop a novel algorithm, which is then implemented in Fortran with message passing interface. Its benefits in terms of runtimes and memory footprint are examined when compared against inverses obtained using the well-known libraries ScaLAPACK and MUMPS.

The implementation scales efficiently to a larger number of cores for larger matrices than previously attempted. The impact of matrix condition number on the accuracy and scalability of the algorithm is evaluated as well.

Fractional diffusion models have numerous applications related to electromagnetic fluid flows, contaminant transport in porous media, material science, image segmentation, etc. Čiegis et al[4] develop and investigate parallel algorithms for state-of-the-art numerical methods of solving diffusion problems with fractional powers of elliptic operators. A two-level parallelization approach is applied to construct efficient parallel algorithms using the domain decomposition and master-slave approach to deal with the increase in computational complexity. The important objective of the research is to analyze and compare the scalability and accuracy of the developed algorithms. The performed analysis shows that selecting the best parallel numerical method for the parallel solution of a problem with fractional power of an elliptic operator is a highly nontrivial task, as it is problem and accuracy dependent.

The efficient usage of heterogeneous systems is infeasible without developing efficient algorithms and software environments with a suitable combination of features to match various forms of parallelism corresponding to different devices. Laccetti et al[5] introduce an adaptive procedure for the numerical computation of high-dimensional integrals on HPC systems with heterogeneous CPU-GPU nodes. To this end, two different approaches are integrated together: The first one is in charge of a fair workload distribution among the threads running on multicore CPUs, while the second approach is responsible for an efficient execution of computational kernels on many-core GPUs. The results of experiments with different test functions show that, in some cases, the developed algorithm allows increasing the performance more than five times compared with a solution using only CPUs. In future work, the authors plan to study the advantages of integrating the proposed methodology with a message passing strategy.

Dynamic programming is an important algorithmic technique to find the optimal solution of a problem over an exponential number of solution candidates. Yamashita et al[6] propose a parallel processing technique for the bulk execution of the dynamic programming on GPUs, assuming many instances of a problem. In particular, they focus on the optimal polygon triangulation problem for a lot of polygons. To perform efficiently, the parallel execution uses a lot of threads on thousands of cores in the GPU, and to accomplish that, a novel task-parallel thread assignment technique is developed addressing the well-known programming issues of the GPU architecture. The experimental results show that the developed GPU implementation on NVIDIA Titan V graphics card for PCs attains a speedup factor of up to 26 times against the eight-thread CPU implementation on Intel Core i7. The proposed technique can be also applied to the bulk execution for other dynamic programming problems.

Due to specific characteristics of GPU computing and memory accessing limitations, algorithms and data structures attempting to harness parallelism enabled by this architecture must be more or less redesigned. Kaczmarski and Wolant[7] present a general-purpose dictionary with parallel database creation and lookup redesigned and tuned for GPU accelerators with ability of customization for different data structures. This dictionary is based on R-Trie as a multiway retrieval method with arbitrary length binary keys. The experiments show that resulting R-Trie dictionary allows achieving a very small creation latency and ultrafast key lookup. The proposed solution proves to be efficient in the most demanding applications such as network routing using the longest prefix match algorithm and multiple pattern matching. At the same time, the authors identified several challenges regarding memory consumption, which they are planning to be addressed in future works.

Discrete-event simulation (DES) is a well-known technique used for the modeling and simulation of complex systems over different timescales. Maśko and Tudruj[8] focus on distributed/parallel DES problems, which assume that multiple simulated event queues are processed in parallel using the time warp approach. The authors propose a novel control method for the optimistic parallel event-driven simulation based on asynchronous monitoring of the simulator global states. This method prevents excessive rollbacks in the time warp simulation that could suffocate speedup of the entire simulation. The efficiency of the proposed solution was confirmed in a series of experiments with PHOLD, that is, the most widely used benchmark for performance evaluation of parallel DES systems.

First-principle calculations of the electronic structure have been one of the most important classes of supercomputer applications for a long time. VASP is the de facto standard tool for density functional theory calculations widely applied in materials science, condensed matter research, and related fields. The wide spectrum of hardware options is the reason why the choice of the best variant for a particular workload pattern is rather complicated. Using a typical VASP workload example, Stegailov et al[9] present the efficiency analysis for a limited but representative list of modern 64-bit CPUs and Nvidia GPUs. They consider both time-to-solution and energy-to-solution criteria. In particular, the CUDA version of VASP with 2 Nvidia Tesla P100 GPUs per node shows very competitive performance and the best combination of a time-to-solution with an energy-to-solution. However, further scalability (either to a higher number of GPUs per node or to a higher number of nodes) is not very efficient.

A growing interest toward energy aspects of computation has led to reinvestigation of scheduling algorithms. Rauber and Rünger[10] follow this research direction and study the impact of task scheduling algorithms on the performance and energy consumption of applications. The authors consider the problem of scheduling a set of independent tasks on a parallel system with homogeneous execution units providing frequency scaling. It is assumed that execution time and energy consumption are nonlinear with respect to frequency scaling. To assign tasks to execution units, the authors propose a selection process combining scheduling algorithms, which determine a task assignment, with a subsequent selection of frequency scaling. This process builds a rich set of alternative schedule execution modes from which efficient (Pareto-optimal) execution modes can be selected. The results of experiments performed for the SPEC CPU benchmarks show that the enriched scheduling process leads to task assignments resulting in an efficient execution on dynamic voltage and frequency scaling processors.

ACKNOWLEDGMENTS

The guest editors of this special issue wish to thank the reviewers for the careful evaluation of the papers and useful suggestions that helped the authors to improve their contributions. Also, we would like to sincerely thank Professor Geoffrey Fox (Editor-in-Chief of the *Concurrency and Computation: Practice and Experience*) for the opportunity to edit this special issue and for his guidance during this process.

ORCID

Roman Wyrzykowski https://orcid.org/0000-0003-1724-1786

Roman Wyrzykowski[1]
Boleslaw K. Szymanski[2]
[1]*Czestochowa University of Technology, Czestochowa, Poland*
[2]*Rensselaer Polytechnic Institute, Troy, New York*

Correspondence
Roman Wyrzykowski, Czestochowa University of Technology, Dabrowskiego 69, 42-201 Czestochowa, Poland.
Email: roman@icis.pcz.pl

REFERENCES

1. Krestenitis K, Weinzierl T. A multi-core ready discrete element method with triangles using dynamically adaptive multiscale grids. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
2. Mikkelsen CCK, Schwarz AB, Karlsson L. Parallel robust solution of triangular linear systems. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
3. Spellacy L, Golden D, Rungger I. Performance analysis of a pairwise method for partial inversion of complex block tridiagonal matrices. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
4. Čiegis R, Starikovičius V, Margenov S, Kriauzienė R. Scalability analysis of different parallel solvers for 3D fractional power diffusion problems. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
5. Laccetti G, Lapegna M, Mele V, Montella R. An adaptive algorithm for high-dimensional integrals on heterogeneous CPU-GPU systems. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
6. Yamashita K, Ito Y, Nakano K. Bulk execution of the dynamic programming for the optimal polygon triangulation problem on the GPU. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
7. Kaczmarski K, Wolant A. GPU R-Trie: dictionary with ultra fast lookup. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
8. Maśko Ł, Tudruj M. Application global state monitoring in optimization of parallel event-driven simulation. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
9. Stegailov V, Smirnov G, Vecher V. VASP hits the memory wall: processors efficiency comparison. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.
10. Rauber T, Rünger G. A scheduling selection process for energy-efficient task execution on DVFS processors. *Concurrency Computat Pract Exper*. 2019. Published online ahead of print.