



Parallel computing technologies 2020

Victor Malyshkin^{1,2,3}

Accepted: 5 August 2021 / Published online: 4 October 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Parallel computing technologies enable the solution of large-scale numerical simulation and data processing problems in science and industry. The special issue highlights a number of research fields from a wide spectrum of parallel computing technologies. The range of research problems spans from architecture-based optimizations of specific programs, through new models/algorithms that allow for natural parallel implementation, to high-level programming instruments and tools for organization of computing experiments and data processing workflows. Accumulation and automatic reuse of good solutions, generation of new solutions is a general trend at all levels.

The paper “Parallel programs execution optimization using behavior control in LuNA system” by V. Malyshkin, D. Akhmed-Zaki, and V. Perepelkin describes the execution of LuNA’s programs. The LuNA programming system automatically generates the target program from a recursively countable set of functional terms that represents an application algorithm. This method is not new (see one of the early works of Yury Yanov: Folding method for properties of formal systems resolving/Preprint of the Institute of Applied Mathematics RAS No 11, 1977), but only relatively recently it began to be widely used for generation of hardware and software discrete systems. The quality of generating discrete systems is also gradually improving and is now becoming more or less acceptable.

Generation is a very promising method of parallel/distributed program automatic construction. It is expected that the method will be used more and more for the development of new programming technologies.

The paper “CUDA-powered method for the feature extraction and unsupervised analysis of medical images” by L. Rundo et al. proposes a new method—a particular data processing pipeline—for medical image analysis and describes its implementation with a focus on CUDA-based parallel implementation of two main steps:

✉ Victor Malyshkin
malysh@ssd.sssc.ru

¹ Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

² Novosibirsk State University, Novosibirsk, Russia

³ Novosibirsk State Technical University, Novosibirsk, Russia

Haralick feature extraction and SOM-based (self-organizing maps) unsupervised image pixel clustering.

Often data comes from many sources. I. Kholod et al. in “Parallelization of the self-organized maps algorithm for federated learning on distributed sources” consider processing of data on data sources instead of aggregating data in a processing centre. The authors propose a formal approach to generate distributed implementation of self-organizing maps. The approach enables executing the parallel algorithm on the distributed data sources, taking into account the kind of data distribution on the nodes.

Development of new approaches in data processing and numerical simulation that naturally allow for parallel implementation is another important field of research. Works on cellular automata is a prominent example of such efforts. R. Hoffmann et al. address, in their paper “A cellular automata rule placing a maximal number of dominoes in the square and diamond”, a specific problem in simulation of pattern formation. The authors consider composition of specified “domino” tiles into a pattern with specific quality requirements. A probabilistic cellular automaton is proposed to solve the problem. This creates a possibility to study and build such patterns with parallel computers because the algorithm is inherently parallel. A. Kireeva et al. contributed another paper in the field of cellular automata: “Parallel simulation of drift-diffusion-recombination by cellular automata and global random walk algorithm”. The paper proposes and studies comprehensively two ways to simulate drift-diffusion-recombination processes: a cellular automaton and the global random walk algorithm. It is shown that results obtained with both models are in good agreement with the exact solution of the drift-diffusion-recombination equation. Parallel programs written to implement the proposed approaches demonstrate near-linear scalability. M. Szaban studies the rules for one-dimensional totalistic cellular automaton such that automaton can be used as a pseudo-random number generator. A systematic approach is presented in the paper “Simple method of selecting totalistic rules for pseudorandom number generator based on nonuniform cellular automaton” to analyse such rules and evaluate resulting pseudo-random sequences. Finally, a set of rules is proposed that define an automaton capable of producing high-quality sequences.

F. Luccio and L. Pagli present new developments on the classical problem of “cops” chasing “robbers” on a graph (“Cops and Robber on Grids and Tori Basic Algorithms and their Extension to a Large Number of Cops”). The problem serves as a model in a number of applications such as motion planning and environment control. A new approach to designing efficient chasing algorithms and finding a minimal number of cops on 2D grids, cylinders and tori is proposed. Also, the paper proposes an idea to study how the time of chasing depends on the number of cops. A method to design chasing algorithms for an arbitrary number of cops is proposed and possible “speed-up” of chasing with increase of the cops (acting in parallel) is evaluated.

The paper “Streaming techniques: revealing the natural concurrency of the lattice Boltzmann method” by A. Zakirov et al. is devoted to the problem of efficient implementation of Lattice Boltzmann models on modern computer architectures, particularly, on GPUs. A new scheme of implementation is proposed that prescribes the

rules for representation of data in memory, memory traversal patterns and distributing work among computing threads. The authors show that the arithmetic intensity of the proposed scheme is higher than the arithmetic intensity of other considered schemes.

The paper “Parallel modeling of wildfires using efficient solvers for ill-conditioned linear systems” by O. Bessonov and S. Meradji continues the series of their works on the development of the FireStar3D model and numerical codes for simulation of wildfires on parallel computers. Two parallel solvers (based on the preconditioned conjugate gradient method and on the algebraic multigrid method) for ill-conditioned linear systems are proposed and analysed. The contribution is twofold: the authors study how the methods should be adapted to the particular case of the wildfires model and propose a parallel implementation for the adapted methods.

One of the forms for accumulation of “good” solutions is a software framework. The paper “VGL: a high-performance graph processing framework for the NEC SX-Aurora TSUBASA vector architecture” (Ilya V. Afanasyev et al.) proposes a framework that can be used for rapid implementation of graph processing algorithms on vector computing systems (NEC SX-Aurora TSUBASA was the main target system) using high-level data abstractions (“graph”, “frontier”—sets of vertices) and computational abstractions (“advance”, “compute”, “generate new frontier”, “reduce”) that are manually implemented to take into account peculiarities of vector architectures. Non-trivial implementation details (optimization techniques) are described in the paper. Several graph processing algorithms were developed on the basis of the framework. Their performance was shown competitive to other known implementations on other architectures. Thus, the paper proposes a new technology for development of graph processing parallel applications for a specific architecture.

O. Sukhoroslov (“Toward efficient execution of data-intensive workflows”) considers the problem of workflow scheduling. The objective of the work is to develop scheduling algorithms that schedule data transfers explicitly to reduce network contention. A model for a job scheduling problem that explicitly includes network contention is proposed. The previously proposed contention-aware scheduling algorithm is formally described with this model, modifications of the algorithm are proposed: look-ahead technique, chunked data transfers, data caching, and peer-to-peer data transfers. The proposed algorithms are evaluated by simulation on a set of workflows and compared to the results of the known HEFT algorithm. The results of experiments demonstrate that no algorithm or approach consistently outperforms others in all experiments. However, the proposed algorithms produce better plans in some of the simulated cases, so they can be used together with other approaches in real-life settings.

M. Gorodnichev and D. Lebedev (“Semantic tools for development of high-level interactive applications for supercomputers”) present a new approach, based on the theory of structural synthesis, and a software framework for development of high-level user interfaces to manage computational experiments and data processing on supercomputers. The objective is to help users, who often have to manage their data, codes, and computing jobs on multiple computing resources manually, to organize such actions into meaningful scenarios (workflows) and support these scenarios with high-level interfaces. The framework provides a language for formal specification of

data objects and operations that connect these objects into a network of possible actions. The framework includes an engine to derive and execute required workflows based on such a network. Web GUI components needed to support user-system interaction during workflow execution are also generated. These interfaces can also be enriched with UI components developed outside of the framework and included into the library of components.

The special issue provides although not a comprehensive but rather representative overview of research in parallel computing technologies. We would like to draw the attention of the readers to the problems of automatic accumulation, reuse of efficient solutions, and generations of new solutions on the basis of formal specifications of the problem domains. Here is the future of parallel computing technologies.

Acknowledgements The editorial work was partially funded by the budget project of the ICMMG SB RAS No. 0251-2021-0005.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.