



Parallel optimization using/for multi and many-core high performance computing

Nouredine Melab, Albert Zomaya, Imen Chakroun

► To cite this version:

Nouredine Melab, Albert Zomaya, Imen Chakroun. Parallel optimization using/for multi and many-core high performance computing. *Journal of Parallel and Distributed Computing*, Elsevier, 2018, 112, pp.109 - 110. 10.1016/j.jpdc.2017.11.011 . hal-01924680

HAL Id: hal-01924680

<https://hal.inria.fr/hal-01924680>

Submitted on 16 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parallel Optimization using/for Multi and Many-core High Performance Computing

N. Melab^a, A. Zomaya^b, I. Chakroun^c

^aUniversité Lille 1 / INRIA Lille Nord Europe / CNRS CRISTAL - Lille, France.

^bThe University of Sydney - Sydney, Australia.

^cExasience Life Lab, IMEC Research Center - Leuven, Belgium.

On the path to exascale which is fast approaching, multi- and many-core (co)processors are increasingly becoming key-building blocks of many computing platforms, as evidenced in the Top500 international ranking. On the other hand, plenty of hard problems in a wide range of areas (biology, energy, etc.) are often modeled and tackled using optimization approaches. These approaches include mainly greedy algorithms, exact methods (dynamic programming, Branch-and-X, constraint programming, A^* , etc.) and meta-heuristics (evolutionary algorithms, particle swarm, ant or bee colonies, simulated annealing, Tabu search, etc.). In many research works, on the one hand, optimization techniques are used to face high performance computing (HPC) challenges including hardware design, compiling, scheduling, auto-tuning, etc. On the other hand, optimization problems become increasingly large and complex, forcing the use of parallel computing for their efficient and effective resolution. The design and implementation of parallel optimization methods raise several issues such as load balancing, data locality and placement, fault tolerance, scalability, thread divergence, etc.

This special issue invited high-quality contributions on the joint use of advanced (discrete, continuous or mixed, single or multi/many-objective, static or dynamic, deterministic or stochastic, hybrid) optimization methods and distributed and/or parallel multi/many-core computing, and any related issues. While the submissions were on the topics of the POMCO'2016 workshop organized in conjunction with HPC&S'2016, the call for papers was open to all contributors. We received 17 submissions from several different countries. The submissions were rigorously reviewed by at least three expert external reviewers, and further carefully evaluated by the guest editors. After the review process, only 5 were finally accepted for publication. We thank all the reviewers for their critical and expert help. Below, we provide an overview of the papers appearing in this volume.

In [5], the authors address the irregular combinatorial problem of finding the smallest binary tree size generated by the iterative partitioning of the regular simplex by a longest edge. Solving this problem requires an increasing

computational burden in size (dimension and stopping criterion). Several multi-core algorithms are proposed and compared using TBB and PThread for threaded implementation. The TBB-based approach uses a static number of threads and features task stealing while the Pthread-based one uses a dynamic number of threads and the creation-destruction policy implicitly implementing load balancing. According to the reported experimental results several interesting observations related mainly to memory management and dynamic load balancing can be highlighted. First, the TBB version outperforms its Pthreaded counterpart for the small instances of the problem, but fails to solve the largest one. The Pthread version could solve more instances than TBB more efficiently, especially for the hardest instances. This can be explained by the way in which threads get work units from the search tree. Indeed, Breadth-First search is activated at deeper levels of the tree which results in less memory requirement. In addition, the LLAlloc memory allocator of LockLess Inc has been demonstrated to be the best among multiple experimentally evaluated allocators.

Another irregular combinatorial problem is tackled using parallel exact optimization in [4]. The problem, called Multiple Sequence Assignment (MSA), is a basic operation in bioinformatics highly useful to find out the similarities among a set of sequences. MSA is modeled as a search for the path with minimum cost in a graph and solved using the state-of-the-art A^* algorithm revisited in the paper using in single-core as well as multi-threaded computing for multi-core platforms. The design of a multi-threaded A^* (PA-Star) for solving MSA raises two major challenges: irregular data access patterns and substantial memory requirements. The first challenge is raised using a one-level hash function with locality preserving characteristics allowing to reduce the overhead induced by the communications between threads, thereby increasing the scalability of the algorithm. The memory issue is addressed using out-of-memory computing. Indeed, PA-Star uses two primary lists of nodes, called *OpenList* and *ClosedList*, to explore the search with a best-first strategy. *OpenList* (resp. *ClosedList*) is used to store the nodes to be (resp. already) analyzed. The disk-assisted strategy consists in saving the nodes of *ClosedList* to the

disk when the RAM is being saturated. The PA-Star algorithm has been experimented on multi-core systems using the BALiBASE reference set. The reported results demonstrate the efficiency of the proposed mechanisms and its superiority over the *PFA* – DDD* state-of-the-art algorithm.

The paper [3] is focused on parallel software framework for solving large problems on multi-core systems. The author proposes an extension, called IBobpp, of the Bobpp framework. Bobpp supports the node-based parallel model which is widely used for parallel tree search. In this model, a global shared pool is used to store the nodes generated by the parallel exploration process. The tree nodes are distributed to threads according to the work stealing paradigm. The shared pool becomes a bottleneck when it comes to solve large problem instances limiting thereby the scalability of the framework. To deal with this issue the author proposes in IBobpp another approach using multiple pools. In addition, he proposes an implementation of the tree-based coarse-grained parallel model. In this model, each thread explores locally and serially a sub-tree limiting the amount of memory accesses to the shared pool. Extensive experiments reported in the paper show that the combination of the node-based and tree-based parallel models together with the use of multiple pools allows to improve the performance and scalability of the framework.

The work presented in [1] deals with the parallelization of metaheuristics, which are near-optimal algorithms. More exactly, the focus is put on the Particle Swarm Optimization (PSO) metaheuristic for solving multi-objective optimization problems on multi-core systems. The authors propose a parallel variant of the SMPSO, which stands for Speed-constrained Multi-objective PSO, algorithm. The new algorithm called CCSMPSO, which stands for Cooperative Co-evolutionary SMPSO, is based on the cooperative co-evolution paradigm. This later consists in breaking down the population of the algorithm into multiple sub-populations that evolve specific parts of the global solution by cooperating during the evolution. Extensive experiments have shown that CCSMPSO outperforms the CC versions of state-of-the-art multi-objective metaheuristics, namely NSGA-II, SPEA2 and MOCcell as well as their serial counterparts. In addition, the reported results highlight that CCSMPSO scales better when increasing the size of the problem (number of decision variables) as well as the number of processing cores. In particular, the algorithm improves the quality of obtained Pareto fronts (solutions) while the others decrease it.

In the four previous papers, fast execution is targeted but not energy efficiency. In [2], the focus is put on this later criterion within the context of heterogeneous data centers. More exactly, the authors propose a new detailed

model of a data center considering the power consumption, the performance of the compute nodes and cooling system, thermal constraints, DVFS and co-location interference. This later is crucial in modern data centers including massively multi-core compute nodes. Co-located tasks run on the processing cores within the same processor leading to co-location interference effect because of the use of shared memory and last-level cache. In addition, the authors propose, experiment and rigorously analyze three resource management techniques: a greedy heuristic, a genetic algorithm combined with a local search and a revisited non-linear programming approach. Co-location is considered together with power and thermal constraints.

Finally, we warmly thank all the authors for their high quality contributions. We hope that this volume will be of timely value to the readers.

References

- [1] Arash Atashpendar, Bernabe Dorronsoro, Gregoire Danoy and Pascal Bouvry. A Scalable Parallel Cooperative Coevolutionary PSO Algorithm for Multi-objective Optimization. *Journal of Parallel and Distributed Computing*, 2017.
- [2] Mark Oxley, Eric Jonardi, Sudeep Pasricha, Anthony A. Maciejewski, Howard J. Siegel, Patrick J. Burns and Gregory A. Koenig. Rate-based Thermal, Power, and Co-location Aware Resource Management for Heterogeneous Data Centers. *Journal of Parallel and Distributed Computing*, 2017.
- [3] Tarek Menouer. Solving Combinatorial Problems using a Parallel Framework. *Journal of Parallel and Distributed Computing*, 2017.
- [4] Daniel Sundfeld, Caina Razzolini, George Teodoro, Azzedine Boukerche and Alba Cristina Melo. PA-Star: a Disk-Assisted Parallel A-Star Strategy with Locality-Sensitive Hash for Multiple Sequence Alignment. *Journal of Parallel and Distributed Computing*, 2017.
- [5] Guillermo Aparicio, Jose Manuel G Salmerón, Leocadio Gonzalez González Casado, Rafael Asenjo, Eligius M.T. Hendrix and Leocadio Gonzalez Gonzalez Casado. Parallel algorithms for computing the smallest binary tree size in unit simplex refinement. *Journal of Parallel and Distributed Computing*, 2017.

GUEST EDITORS

- **Nouredine Melab**
 Université Lille 1 / INRIA Lille Nord Europe / CNRS CRISTAL - Lille, France.
 Phone: +33 3 59 57 78 86
 Email: nouredine.melab@univ-lille1.fr
- **Albert Y. Zomaya**
 The University of Sydney - Sydney, Australia.
 Phone: +61-2-9351 6442
 Email: albert.zomaya@sydney.edu.au
- **Imen Chakroun**
 Exascience Life Lab, IMEC Research Center - Leuven, Belgium.
 Phone: +32 16 28 81 89
 Email: imen.chakroun@imec.be