Proposal for a Tutorial at SPAA 2024

Taming the Zoo of Parallel Machine Models

by Peter Sanders
Karlsruhe Institute of Technology, Germany

Abstract. There is an intimidating multitude of models for parallel computing which form the basis for parallel algorithm design. This tutorial gives an overview of the most important models and tries to bring some order into the chaos. Why are there so many different models? Which one should I use? What are simple relations between the models? When do differences really matter? When can we use a model-agnostic approach that is largely independent on modeling details? What are open problems? All this will be considered from the point of view of algorithm engineering, i.e., taking both theory and practice into account.

Content. Starting from the standard sequential RAM model, we will introduce different variants of PRAMs leading to more realistic asynchronous shared-memory models with memory hierarchies. We then look at distributed-memory models like BSP and point-to-point message exchange.

We contrast this to more abstract models like MapReduce and distributed graph computations (e.g., LOCAL and CONGEST) as well as communication complexity.

Then we look at relations between models in terms of one model emulating the other. Depending on the available time, further models like circuits, cellular automata, or quantum computing will be considered. A final attempt at a synthesis stresses that a large number of these models have a valid role but none should be overestimated. On the other hand, a small number of models suffices for many uses and often we can even get by with a model-agnostic view where we describe algorithms in terms of basic primitives that have concrete implementations in many models.

Simple warm-up exercises will ask for translating pseudo-codes for basic algorithms like broadcast from one model to another. Further exercises will ask questions on developing and analyzing algorithms for emulating one model on the other. These range from simple to open research problems. Overall, participants with different levels of experience can pick something adequate.

The tutorial is based on a ca. 100-page chapter on machine models in the tutor's future book on algorithm engineering. A draft version of the chapter will be available for the tutorial.

Prerequisites. The Tutorial assumes knowledge on the level of a master's degree in computer science and some previous experience with parallel computing and parallel algorithms. Most of the concrete previous knowledge we assume will already be covered by the algorithmic courses of a bachelor's degree.

Biography. Peter Sanders received his PhD in computer science from Universität Karlsruhe in 1996. After a short PostDoc at Chalmers University, he joined Max-Planck-Institute for Informatics in Saarbrücken. After 7 years, he returned to Karlsruhe as a full professor in

2004. He works at the Informatics Department of KIT in the Institute for Theoretical Computer Science. Peter Sanders won several prices among them the DFG Leibniz Award 2012 and a 2020 ERC advanced grant for the project "Engineering Scalable Algorithms for the Basic Toolbox – ScAlBox". He has more than 250 publications, mostly on algorithms for large data sets. This includes parallel algorithms (sorting, data structures, multi-core algorithm libraries, load balancing, communication-efficient algorithms, memory hierarchies, graph algorithms, randomized algorithms, full-text indices, etc. He is very active in promoting the methodology of algorithm engineering that integrates design, analysis, implementation, and experimental evaluation of algorithms.