Keynote Talk: Parallel Algorithm Engineering Reconsidered

Peter Sanders
sanders@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Baden Württemberg, Germany

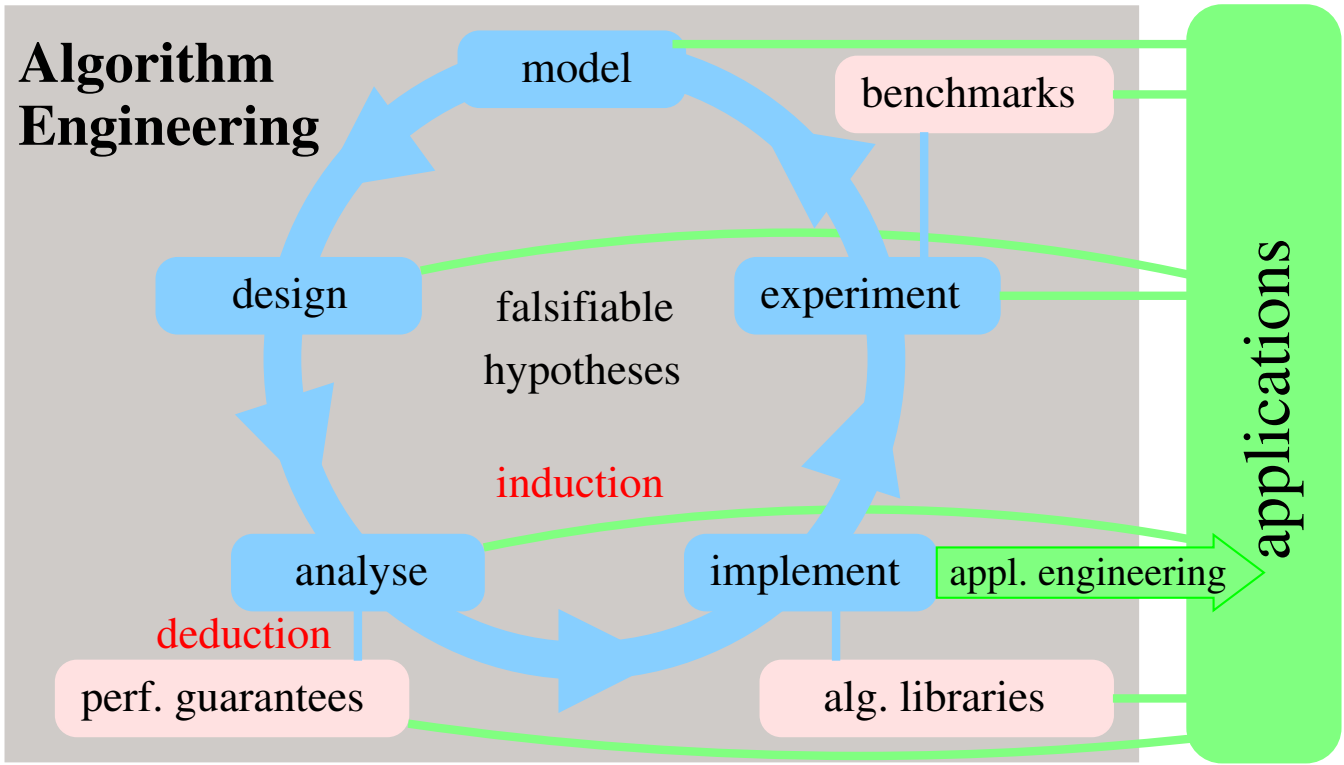


Figure 1: The algorithm engineering cycle

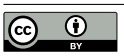
Abstract

Algorithm engineering is a view on the methodology of algorithmic research. At its core is a feedback cycle of modeling, design, analysis, implementation, and experimental evaluation [1, 3]; see Fig. 1. At each stage, we learn new things about an algorithmic problem that drives the development of better solutions. This is inspired by the hypothesis-driven cycle of scientific discovery prevalent in the natural sciences [2].

The invited talk reflects on the role of algorithm engineering with particular focus on parallel computing. Modeling is particularly important there because no standard model has emerged after more than half a century of research on parallel algorithms. For example, MapReduce computations have become a popular high-level theoretical model but their connection to the real world is an interesting question [4]. On the low-level end of this spectrum, we need models that enable extensive algorithm-hardware codesign needed for future energy-efficient AI hardware.

On the design side, it emerges that we do not just need a single algorithm but rather a spectrum of solutions selected from a huge design space of possible algorithms. This spectrum can then cover a range of inputs and machines. To grasp this on the analysis side, we need to take constant factors into account and handle complex models. Implementation has moved from a simple transition stage to extensive performance engineering to take various kinds of parallelism into account, e.g., bits, SIMD, instructions, threads in hierarchies of caches, and multipe compute nodes. This can result in orders of magnitude performance differences, often outstripping asymptotic behavior.

Experiments and the entire cycle are now in a revolutionary stage where AI assisted algorithm engineering can drive the cycle



PODC '26, Egham, United Kingdom

ACM ISBN 979-8-4007-2512-8/26/07
https://doi.org/10.1145/3796701.3820440

at unprecedented speed. As human researchers, we rapidly have to learn how to control it.

CCS Concepts

• **Computing methodologies → Parallel computing methodologies**; • **Theory of computation → Design and analysis of algorithms**.

Keywords

algorithm engineering, machine model, algorithm design, algorithm analysis, performance engineering, experimental algorithmics, AI assisted algorithm engineering, algorithm hardware codesign

ACM Reference Format:
Peter Sanders. 2026. Keynote Talk: Parallel Algorithm Engineering Reconsidered. In *ACM Symposium on Principles of Distributed Computing (PODC '26), July 6–10, 2026, Egham, United Kingdom.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3796701.3820440

Short Biography

Peter Sanders received his PhD in Computer Science from Universität Karlsruhe in 1996. After a short postdoctoral period at Chalmers University in Gothenburg, Sweden, he joined the Max Planck Institute for Informatics in Saarbrücken. In 2004, he returned to Karlsruhe as a full professor. He currently leads the Algorithm Engineering Group in the Institute for Theoretical Computer Science at the Karlsruhe Institute of Technology (KIT). He also works part-time for Google.

Professor Sanders has received numerous honors and awards, including the DFG Leibniz Award in 2012 and a 2020 ERC Advanced Grant for the project "Engineering Scalable Algorithms for the Basic Toolbox (ScAlBox)". He has authored more than 300 publications, primarily in the area of algorithms for large-scale data processing.

His research spans a broad range of topics, including parallel algorithms (such as sorting, SAT solving, data structures, multicore algorithm libraries, load balancing, and communication scheduling), memory hierarchies (caches, disks, and storage servers), graph algorithms (including route planning and graph partitioning), randomized algorithms, compressed data structures, and full-text indexing.

References

[1] D. Bader, B. Moret, and P. Sanders. 2002. Algorithm Engineering for Parallel Computation. In *Experimental Algorithmics — From Algorithm Design to Robust and Efficient Software.* LNCS, Vol. 2547. Springer, 1–23.
[2] K. R. Popper. 1934. *Logik der Forschung.* Springer. English Translation: *The Logic of Scientific Discovery*, Hutchinson, 1959.
[3] P. Sanders. 2009. Algorithm Engineering – An Attempt at a Definition. In *Efficient Algorithms.* LNCS, Vol. 5760. Springer, 321–340.
[4] Peter Sanders. 2020. Connecting MapReduce Computations to Realistic Machine Models. In *IEEE Conference on Big Data.* 84–93. https://doi.org/10.1109/BigData50022.2020.9378039