

Invited Talk: Some Sequential Algorithms are Almost Always Parallel

Guy E. Blelloch
Carnegie Mellon University
Pittsburgh, PA, USA
guyb@cs.cmu.edu

ABSTRACT

Over the years many interesting and efficient parallel algorithms have been developed to solve a wide variety of problems, but not much attention has been paid to studying the inherent parallelism in sequential algorithms—i.e., understanding the depth of their dependence structure, and how shallow dependence structures might be used to develop efficient parallel implementations.

In this talk I will describe recent work on analyzing the dependence depth of iterative sequential algorithms—ones that loop over a collection of elements. Many of these algorithms have deep dependence chains in the worst case, but shallow chains (polylog w.h.p.) if the elements are randomly ordered. Examples include many fundamental algorithms: the Knuth shuffle for random permutations, sorting by insertion into a binary search tree, greedy maximal independent set (MIS), greedy maximal matching, greedy graph-coloring, counting cycles in a permutation, incremental k-dimensional linear programming, and incremental 2d Delaunay triangulation. An advantage of the approach is that it can lead to very simple and efficient parallel algorithms. Our MIS algorithm, for example can be coded in a dozen or so lines, and is significantly faster than Luby's algorithm on modern multicore machines. Also the approach encourages snapping the view that sequential and parallel algorithms are distinct, and instead thinking of algorithms, in general, as collections of instructions with dependences among them

BIOGRAPHY

Guy Blelloch is a Professor of Computer Science at Carnegie Mellon University. He received a BA in Physics and BS in Engineering from Swarthmore College in 1983, and a PhD in Computer Science from MIT in 1988. His research interests are in algorithms and programming languages and how they interact, with an emphasis on parallel computation. He has worked on programming-language based cost models along with provable implementation bounds, on bounding costs in runtime scheduling and parallel garbage collection, and designed the Nesl programming language. He has developed a variety of parallel algorithms, including algorithms for graph connectivity, set cover, semi sorting, suffix trees, balanced trees, hashing, shortest paths, and Delaunay triangulation. His work has both been in models that are communication conscious, e.g. accounting for caching, and not. He is most proud of developing a sophomore level algorithms and data structures course at Carnegie Mellon in which parallelism is taught from the start. He is an ACM Fellow for his contributions in parallel computation, and was general chair of the ACM Symposium on Parallelism in Algorithms and Architecture (SPAA) from 2011–2015.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PODC'17 & SPAA'17, July 24-27, 2017, Washington, DC, USA.

© 2017 Copyright is held by the author/owner(s).

ACM ISBN 978-1-4503-4992-5/17/07 & 978-1-4503-4593-4/17/07.

DOI: <http://dx.doi.org/10.1145/3087801.3087871>