

**32nd ACM Symposium on
PARALLELISM IN ALGORITHMS AND ARCHITECTURES (SPAA)
TUTORIALS
14 July, 2020**

<https://spaa.acm.org>

All time in EDT (GMT-4), where the conference was originally located. Convert time zones: CET: +6h, CDT: -1h, PDT: -3h.

1:00 – 4:00 pm: Research and Teaching with OpenCilk

By: Dorothy Curtis (MIT), I-Ting Angelina Lee (WUSTL), Alexandros-Stavros Iliopoulos (MIT), Charles E. Leiserson (MIT), Tao B. Schardl (MIT)

Abstract:

OpenCilk is a new open-source platform to support Cilk task-parallel programming in C/C++, aimed primarily at researchers and teachers of parallel computing. OpenCilk provides a full-featured implementation of Cilk, including a compiler based on Tapir/LLVM and a runtime system based on Cheetah, as well as integrated tools, including a race detector and a scalability analyzer. OpenCilk provides all the task-parallel features of Intel Cilk Plus. This tutorial should facilitate existing users of Cilk Plus to migrate to OpenCilk, and some familiarity with the Cilk language will be assumed.

The first half of the tutorial overviews the OpenCilk project, provides a look under the hood of the OpenCilk architecture, and describes the components and features. We will also discuss future development and solicit input from the audience about development priorities. The second half of the tutorial will be a hands-on session, where you can download a beta version of OpenCilk for Linux, BSD, and Mac OS X to experiment with running your existing Cilk codes on the new platform. Live one-on-one assistance will be provided through the SPAA2020 Slack channel.

4:00 – 5:30 pm: Implementing Parallel Tree Structure in Shared-Memory

By: Yihan Sun (UC Riverside)

Abstract:

This tutorial will introduce a simple and efficient parallel tree data structure, P-Trees, along with some general techniques about supporting parallelism and concurrency on trees. The core in P-Trees is a framework for parallel balanced binary trees, which bases all tree algorithms on a single primitive "Join". This framework is extendable to multiple balancing schemes. Based on this framework, this tutorial will address techniques including a wide range of algorithms, concurrency, an augmentation framework, and multi-versioning. All algorithms on trees are theoretically efficient. The algorithms are implemented in an open-source C++ library called PAM.

This tutorial will also discuss how to use the algorithms and the PAM library to solve real-world problems, including 2D range/segment/rectangle search, inverted index searching, HTAP database systems, multi-version concurrency control, graph processing systems, etc. Making use of the library, all applications have concise code to get high-performance implementations.

OpenCilk development is supported in part by the National Science Foundation under Grant No. CNS-1925609. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the presenters and do not necessarily reflect the views of the National Science Foundation.

OpenCilk development is supported in part by the United States Air Force Research Laboratory and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this material are those of the presenters and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute content for Government purposes notwithstanding any copyright notation herein.