

Tao B. Schardl

MIT Stata Center, 32-G766
32 Vassar Street
Cambridge, MA 02139 USA

neboat@mit.edu
<http://neboat.mit.edu/>
Updated August 22, 2025

Short biography

Tao B. (TB) Schardl is a Research Scientist in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT and Chief Architect of the OpenCilk task-parallel programming platform. His research aims to make software performance engineering a viable replacement for Moore's Law. To this end, his research integrates algorithms with systems and spans the areas of parallel programming models, theories of software performance, compilers, runtime systems, diagnostic tools, parallel algorithms, and the future of computer performance. He received the US Department of the Air Force Artificial Intelligence Accelerator Scientific Excellence Award in 2022 for his work on OpenCilk. His work on the Tapir/LLVM compiler received the best paper award at the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP) in 2017. His work on computer performance in the post-Moore's Law era was published in Science and has been spotlighted in two Turing-award lectures. Dr. Schardl received his S.B. and M.Eng. in Computer Science and Electrical Engineering from MIT in 2009 and 2010, respectively, and his Ph.D. in Computer Science and Engineering from MIT in 2016.

Citizenship

U.S. Citizen

Education

Ph.D. in Computer Science and Engineering	September 2016
Massachusetts Institute of Technology	Cambridge, MA
<i>Thesis:</i> Performance Engineering of Multicore Software: Developing a Science of Fast Code for the Post-Moore Era	
<i>Advisor:</i> Professor Charles E. Leiserson	
Master of Engineering in Computer Science and Electrical Engineering	June 2010
Massachusetts Institute of Technology	Cambridge, MA
<i>Thesis:</i> Design and Analysis of a Nondeterministic Parallel Breadth-First Search Algorithm	
<i>Advisor:</i> Professor Charles E. Leiserson	
Bachelor of Science in Computer Science and Electrical Engineering	June 2009
Massachusetts Institute of Technology	Cambridge, MA
GPA: 4.9/5.0	

Research experience

Research scientist 3	MIT CSAIL	July 2017–present
<i>PI:</i> Professor Charles E. Leiserson	Supertech Research Group	
	Cambridge, MA	
Postdoctoral associate	MIT CSAIL	September 2016–June 2017
<i>PI:</i> Professor Charles E. Leiserson	Supertech Research Group	
	Cambridge, MA	

Research assistant <i>Advisor: Professor Charles E. Leiserson</i>	MIT CSAIL Supertech Research Group Cambridge, MA	August 2010–August 2016
Intern	U.S. Department of Defense	Summer 2008, Summer 2009

Researched methods for comparing algorithmic differences between two version of a function in a computer program.

Teaching experience

Instructor	6.172: Performance Engineering of Software Systems (U) MIT EECS	Fall 2019
[6.7/7.0 overall rating]		
<i>Course page: https://learning-modules.mit.edu/class/index.html?uuid=/course/6/fa19/6.172</i>		
Instructor	6.172/6.871: Performance Engineering of Software Systems (U/G) MIT EECS	Fall 2017
[6.8/7.0 overall rating; Awarded MIT EECS Department Outstanding Educator Award]		
<i>Course page: https://learning-modules.mit.edu/class/index.html?uuid=/course/6/fa17/6.172</i>		
Instructor	6.S898: Advanced Performance Engineering for Multicore Applications (G) MIT EECS	Spring 2017
Assistant facilitator	6.886: Advanced Performance Engineering for Multicore Applications (G) MIT EECS	Spring 2015
Teaching assistant	6.172: Performance Engineering of Software Systems (U) MIT EECS	Fall 2014
[6.8/7.0 overall rating]		
<i>Course page: http://stellar.mit.edu/S/course/6/fa14/6.172/index.html</i>		
Lecture scribe	6.172: Performance Engineering of Software Systems (U) MIT EECS	Fall 2011
<i>Course page: http://stellar.mit.edu/S/course/6/fa11/6.172/index.html</i>		
Teaching assistant	6.046: Design and Analysis of Algorithms (U) MIT EECS	Fall 2009
<i>Course page: http://stellar.mit.edu/S/course/6/fa09/6.046/index.html</i>		

Awards and honors

Keynote at the 14th International Workshop on Programming Models and Applications for Multicores and Manycores <i>OpenCilk: Architecting a Task-Parallel Software Infrastructure for Modularity, Extensibility, and Performance</i>	February 2023
United States Department of the Air Force Artificial Intelligence Accelerator Scientific Excellence Award <i>For architecting OpenCilk, including inventing and implementing numerous innovative software mechanisms incorporated within this modular and fully open-source task-parallel programming platform.</i>	
Best Paper Award Finalist <i>Received from APoCS, 2020 for “Cilkmem: Algorithms for Analyzing the Memory High-Water Mark of Fork-Join Parallel Programs.”</i>	January 2020
MIT EECS Department Outstanding Educator Award	
Best Paper Award <i>Received at PPoPP, 2017 for “Tapir: Embedding Fork-Join Parallelism into LLVM’s Intermediate Representation.”</i>	February 2017
Akamai Fellowship	2015

Outstanding Paper Award <i>Received from JIP, 2013 for "Finding a Hamiltonian Path in a Cube with Specified Turns is Hard."</i>	June 2014
NSF Graduate Research Fellowship <i>Received from National Science Foundation.</i>	2010–2015
Charles and Jennifer Johnson CS M.Eng. Prize <i>Received for M.Eng. thesis on a work-efficient parallel breadth-first search algorithm.</i>	May 2010
Siebel Scholar <i>Received from Siebel Foundation.</i>	2009–2010
Robert M. Fano UROP Award for Outstanding EECS UROP <i>Received for work on a work-efficient parallel breadth-first search algorithm.</i>	May 2009
Arnold L. Nylander Advanced Undergraduate Project Award <i>Received for work on a work-efficient parallel breadth-first search algorithm.</i>	May 2009
Northern Telecom/BNR Project Award for Best 6.111 Laboratory Project <i>Received for project on voice recognition in hardware.</i>	May 2009
Stokes Educational Scholarship Program <i>Received from U.S. Department of Defense.</i>	2005–2009

Society memberships

IEEE (<i>Member</i>)	2015–present
SIAM (<i>Member</i>)	2012–present
ACM (<i>Member</i>)	2010–present
Phi Beta Kappa National Honor Society (<i>Member</i>)	2009–present
Sigma Xi Scientific Research Society (<i>Associate Member</i>)	2009–present

Publications

Aaron Handelman, Kyle Singer, Tao B. Schardl, and I-Ting Angelina Lee. “Towards Zero Spawn Overhead: Work Stealing Without Deques”. In: *SPAA*. 2025, pp. 75–88. doi: 10.1145/3694906.3743349.

Tim Kaler, Xuhao Chen, Brian Wheatman, Dorothy Curtis, Bruce Hoppe, Tao B. Schardl, and Charles E. Leiserson. “Speedcode: Software Performance Engineering Education via the Coding of Didactic Exercises”. In: *EduPar*. 2024, pp. 391–394. doi: 10.1109/IPDPSW63119.2024.00087.

Helen Xu, Tao B. Schardl, Michael Pellauer, and Joel S. Emer. “Optimizing Compression Schemes for Parallel Sparse Tensor Algebra”. In: *HPEC*. 2023, pp. 1–7. doi: 10.1109/HPEC58863.2023.10363624.

Tim Kaler, Alexandros-Stavros Iliopoulos, Philip Murzynowski, Tao B. Schardl, Charles E. Leiserson, and Jie Chen. “Communication-Efficient Graph Neural Networks with Probabilistic Neighborhood Expansion Analysis and Caching”. In: *MLSys*. 2023. URL: https://proceedings.mlsys.org/paper_files/paper/2023.

Tao B. Schardl and I-Ting Angelina Lee. “OpenCilk: A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code”. In: *PPoPP*. 2023, pp. 189–203. doi: 10.1145/3572848.3577509.

Rocío Carratalá-Sáez, Arturo González-Escribano, Alexandros-Stavros Iliopoulos, Charles E. Leiserson, Charlotte Park, Isabel Rosa, Tao B. Schardl, Yuri Torres, and David P. Bunde. “Peachy Parallel Assignments”. In: *EduHPC*. 2022, pp. 50–56. doi: 10.1109/EduHPC56719.2022.00012.

Tim Kaler, Nickolas Stathas, Anne Ouyang, Alexandros-Stavros Iliopoulos, Tao B. Schardl, Charles E. Leiserson, and Jie Chen. “Accelerating Training and Inference of Graph Neural Networks with Fast Sampling and Pipelining”. In: *MLSys*. 2022. URL: https://proceedings.mlsys.org/paper_files/paper/2022.

Yifan Xu, Anchengcheng Zhou, Grace Q. Yin, Kunal Agrawal, I-Ting Angelina Lee, and Tao B. Schardl. “Efficient Access History for Race Detection”. In: *ALENEX*. 2022, pp. 117–130. doi: 10.1137/1.9781611977042.10.

Charles E. Leiserson and Tao B. Schardl. "A Work-Efficient Parallel Breadth-First Search Algorithm (or How To Cope With the Nondeterminism of Reducers)". In: *Massive Graph Analytics*. Ed. by David A. Bader. 2022, pp. 3–33. doi: 10.1201/9781003033707-2.

William Hasenplaugh, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. "Ordering Heuristics for Parallel Graph Coloring". In: *Massive Graph Analytics*. Ed. by David A. Bader. 2022, pp. 193–221. doi: 10.1201/9781003033707-11.

Tim Kaler, William Hasenplaugh, Tao B. Schardl, and Charles E. Leiserson. "Executing Dynamic Data-Graph Computations Deterministically Using Chromatic Scheduling". In: *Massive Graph Analytics*. Ed. by David A. Bader. 2022, pp. 397–429. doi: 10.1201/9781003033707-18.

Aaron Handleman, Arthur G. Rattew, I-Ting Angelina Lee, and Tao B. Schardl. "A Hybrid Scheduling Scheme for Parallel Loops". In: *IPDPS*. 2021, pp. 587–598. doi: 10.1109/IPDPS49936.2021.00067.

Tim Kaler, Tao B. Schardl, Brian Xie, Charles E. Leiserson, Jie Chen, Aldo Pareja, and Georgios Kollias. "PARAD: A Work-Efficient Parallel Algorithm for Reverse-Mode Automatic Differentiation". In: *APoCS*. 2021, pp. 144–158. doi: 10.1137/1.9781611976489.11.

Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez, and Tao B. Schardl. "There's plenty of room at the Top: What will drive computer performance after Moore's law?" In: *Science* 368.6495 (2020). issn: 0036-8075. doi: 10.1126/science.aam9744.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. "EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs". In: *AAAI*. 2020, pp. 5363–5370. doi: 10.1609/aaai.v34i04.5984.

Tim Kaler, William Kuszmaul, Tao B. Schardl, and Daniele Vettorel. "Cilkmem: Algorithms for Analyzing the Memory High-Water Mark of Fork-Join Parallel Programs". In: *APoCS*. 2020, pp. 162–176. doi: 10.1137/1.9781611976021.12.

[**Best paper finalist**].

Tao B. Schardl, William S. Moses, and Charles E. Leiserson. "Tapir: Embedding Recursive Fork-Join Parallelism into LLVM's Intermediate Representation". In: *ACM Transactions on Parallel Computing* 6.4 (Dec. 2019). doi: 10.1145/3365655.

Tao B. Schardl and Siddharth Samsi. "TapirXLA: Embedding Fork-Join Parallelism into the XLA Compiler in TensorFlow Using Tapir". In: *HPEC*. Sept. 2019, pp. 1–8. doi: 10.1109/HPEC.2019.8916312.

I-Ting Angelina Lee and Tao B. Schardl. "Efficient Race Detection for Reducer Hyperobjects". In: *ACM Trans. Parallel Comput.* 4.4 (Aug. 2018). issn: 2329-4949. doi: 10.1145/3205914.

Tao B. Schardl, I-Ting Angelina Lee, and Charles E. Leiserson. "Brief Announcement: Open Cilk". In: *SPAA*. 2018, pp. 351–353. doi: 10.1145/3210377.3210658.

Tao B. Schardl, Tyler Denniston, Damon Doucet, Bradley C. Kuszmaul, I-Ting Angelina Lee, and Charles E. Leiserson. "The CSI Framework for Compiler-Inserted Program Instrumentation". In: *Abstracts of SIGMETRICS*. 2018, pp. 100–102. doi: 10.1145/3219617.3219657.

Tao B. Schardl, Tyler Denniston, Damon Doucet, Bradley C. Kuszmaul, I-Ting Angelina Lee, and Charles E. Leiserson. "The CSI Framework for Compiler-Inserted Program Instrumentation". In: *SIGMETRICS* 1.2 (Dec. 2017), 43:1–43:25. doi: 10.1145/3154502.

Tao B. Schardl, William S. Moses, and Charles E. Leiserson. "Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation". In: *PPoPP*. 2017, pp. 249–265. doi: 10.1145/3018743.3018758.

[**Won best paper award; invited to a special issue of ACM Transactions on Parallel Computing**].

Tao B. Schardl. "Performance Engineering of Multicore Software: Developing a Science of Fast Code for the Post-Moore Era". PhD thesis. Cambridge, MA: Massachusetts Institute of Technology, Sept. 2016. doi: 1721.1/107290.

Tim Kaler, William Hasenplaugh, Tao B. Schardl, and Charles E. Leiserson. "Executing dynamic data-graph computations deterministically using chromatic scheduling". In: *ACM Transactions on Parallel Computing* 3.1 (July 2016), 2:1–2:31. doi: 10.1145/2896850.

Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B. Schardl. "Who Needs Crossings? Hardness of Plane Graph Rigidity". In: *SoCG*. 2016, 3:1–3:15. doi: 10.4230/LIPIcs.SoCG.2016.3.

Charles E. Leiserson, Tao B. Schardl, and Warut Suksompong. "Upper bounds on number of steals in rooted trees". In: *Theory of Computing Systems* 58.2 (Feb. 2016), pp. 223–240. doi: 10.1007/s00224-015-9613-9.

Warut Suksompong, Charles E. Leiserson, and Tao B. Schardl. "On the efficiency of localized work stealing". In: *Information Processing Letters* 116.2 (Feb. 2016), pp. 100–106. doi: 10.1016/j.ipl.2015.10.002.

I-Ting Angelina Lee, Charles E. Leiserson, Tao B. Schardl, Zhunping Zhang, and Jim Sukha. "On-the-fly pipeline parallelism". In: *ACM Transactions on Parallel Computing* 2.3 (Oct. 2015), 17:1–17:42. doi: 10.1145/2809808.

I-Ting Angelina Lee and Tao B. Schardl. "Efficiently detecting races in Cilk programs that use reducer hyperobjects". In: *SPAA*. 2015, pp. 111–122. doi: 10.1145/2755573.2755599.

[Invited to a special issue of *ACM Transactions on Parallel Computing*].

Tao B. Schardl, Bradley C. Kuszmaul, I-Ting Angelina Lee, William M. Leiserson, and Charles E. Leiserson. "The Cilkprof scalability profiler". In: *SPAA*. 2015, pp. 89–100. doi: 10.1145/2755573.2755603.

William Hasenplaugh, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. "Ordering heuristics for parallel graph coloring". In: *SPAA*. 2014, pp. 166–177. doi: 10.1145/2612669.2612697.

Tim Kaler, William Hasenplaugh, Tao B. Schardl, and Charles E. Leiserson. "Executing dynamic data-graph computations deterministically using chromatic scheduling". In: *SPAA*. 2014, pp. 154–165. doi: 10.1145/2612669.2612673.

[Invited to a special issue of *ACM Transactions on Parallel Computing*].

Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B. Schardl. "Finding a Hamiltonian path in a cube with specified turns is hard". In: *Journal of Information Processing* 21.3 (2013), pp. 368–377. doi: 10.2197/ipsjjip.21.368.

[Won outstanding paper award].

Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, Tao B. Schardl, and Isaac Shapiro-Ellowitz. "Folding equilateral plane graphs". In: *International Journal of Computational Geometry & Applications* 23.02 (2013), pp. 75–92. doi: 10.1142/S0218195913600017.

I-Ting Angelina Lee, Charles E. Leiserson, Tao B. Schardl, Jim Sukha, and Zhunping Zhang. "On-the-fly pipeline parallelism". In: *SPAA*. 2013, pp. 140–151. doi: 10.1145/2486159.2486174.

[Invited to a special issue of *ACM Transactions on Parallel Computing*].

Charles E. Leiserson, Tao B. Schardl, and Jim Sukha. "Deterministic parallel random-number generation for dynamic-multithreading platforms". In: *PPoPP*. 2012, pp. 193–204. doi: 10.1145/2145816.2145841.

Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, Tao B. Schardl, and Isaac Shapiro-Ellowitz. "Folding equilateral plane graphs". In: *ISAAC*. 2011, pp. 574–583. doi: 10.1007/978-3-642-25591-5_59.

Charles E. Leiserson and Tao B. Schardl. "A work-efficient parallel breadth-first search algorithm (or how to cope with the nondeterminism of reducers)". In: *SPAA*. 2010, pp. 303–314. doi: 10.1145/1810479.1810534.

Tao B. Schardl. "Design and analysis of a nondeterministic parallel breadth-first search algorithm". MA thesis. Cambridge, MA: Massachusetts Institute of Technology, May 2010. doi: 1721.1/61575.

[Awarded the Charles and Jennifer Johnson CS M.Eng. Prize].

Mentoring and Supervision

Research advisees

Ryan Deng	PhD	MIT EECS	Current
Kenny Zhang	PhD	MIT EECS	Current
Elie Cuevas	MEng	MIT EECS	May 2025
<i>Thesis:</i> Modeling Recursion with Iteration: Enabling LLVM Loop Optimizations for Recursive Data Structure Traversal			
Sabiyyah Ali	MEng	MIT EECS	February 2025
<i>Thesis:</i> Design and Implementation of a Nonblocking Randomized Work Stealing Scheduler			
Satya Holla	MEng	MIT EECS	August 2024
<i>Thesis:</i> Labeling Schemes for Improving CilkSan Performance			
Jay Hilton	MEng	MIT EECS	May 2024
<i>Thesis:</i> Enabling the Rust Compiler to Reason about Fork/Join Parallelism via Tapir			
Luka Govedič	MEng	MIT EECS	June 2023
<i>Thesis:</i> Improving the Performance of Parallel Loops in OpenCilk			
August Trollback	MEng	MIT EECS	February 2023
<i>Thesis:</i> Continuation Stealing in Julia			
Nikhil Reddy	MEng	MIT EECS	September 2022
<i>Thesis:</i> Optimizing Parallel Performance with Work and Span in the OpenCilk Compiler			
Isabel Rosa	MEng	MIT EECS	May 2022
<i>Thesis:</i> Performance Engineering of Directional Message-Passing Algorithms Through a Stencil-Based Approach for Applications in Molecular Dynamics			
Helen Xu	PhD Reader	MIT EECS	February 2022
<i>Thesis:</i> The Locality-First Strategy for Developing Efficient Multicore Algorithms			
Tim Kralj	MEng	MIT EECS	June 2021
<i>Thesis:</i> Composing Parallel Runtime Systems: A Case Study in How to Compose the Julia and OpenCilk Runtimes			
Helen He	MEng	MIT EECS	June 2021
<i>Thesis:</i> Performance Engineering of Reactive Molecular Dynamics Simulations			
Tim Kaler	PhD Reader	MIT EECS	September 2020
<i>Thesis:</i> Programming Technologies for Engineering Quality Multicore Code			
Sev Kozak	MEng	MIT EECS	June 2020
<i>Thesis:</i> Chasing Zero Variability in Software Performance			
Grace Yin	MEng	MIT EECS	May 2020
<i>Thesis:</i> Parallel Exception Handling in Cilk			
Stephanie Ren	MEng	MIT EECS	June 2019
<i>Thesis:</i> Vector-Aware Space Cuts in Stencil Computations			
Nipun Pitimanaaree	MEng	MIT EECS	June 2019

Thesis: Provably Efficient Randomized Work Stealing with First-Class Parallel Loops

Michael Shah PhD Reader Tufts Computer Science August 2017

Thesis: Understanding and Tuning the Performance of Critical Sections with Program Analysis and Software Visualization Tools

William S. Moses MEng MIT EECS June 2017

Thesis: How Should Compilers Represent Fork-Join Parallelism?

Postdocs

Kyle Singer	MIT CSAIL	July 2023–present
Tim Kaler	MIT CSAIL	September 2020–August 2023
Alexandros-Stavros Iliopoulos	MIT CSAIL	June 2020–June 2023

Grants

Modernizing Compiler Design for Platform and Performance Portability Los Alamos National Laboratory	\$1,000,000	Research scientist	August 2024–July 2029
POSE: Phase II: Open Source Ecosystem for OpenCilk National Science Foundation	\$1,500,000	Research scientist	August 2024–July 2026
POSE: Phase I: Open Source Ecosystem for OpenCilk National Science Foundation	\$ 300,000	Research scientist	September 2023–May 2024
CESMIX: Center for the Exascale Simulation of Material Interfaces in Extreme Environments U.S. Department of Energy	\$8,550,000	Research scientist	September 2020–September 2025
Fast AI: Quick Development of Portable High-Performance AI Applications MIT and U.S. Air Force	\$6,050,000	Research scientist	November 2019–September 2024
CCRI: Medium: Cilk Infrastructure for Next-Generation Parallel-Programming Research National Science Foundation	\$1,500,000	Chief architect	September 2019–September 2023
xGraph: Accelerated and Explainable Graph Deep Learning with Applications to Financial Services MIT and IBM	\$ 750,000	Research scientist	September 2019–August 2023
Analysis and Optimization of Parallel Unstructured-Mesh Computations Los Alamos National Laboratory	\$ 600,000	Research scientist	January 2019–September 2023

Software

OpenCilk <https://www.opencilk.org/>, <https://github.com/OpenCilk>
The latest, open-source implementation of the Cilk parallel-computing platform.

fccode <https://www.overleaf.com/read/gbqhfyngcbgy>
L^AT_EX package and Pygments plugin for fast and flexible syntax-highlighting of code.

Tapir/LLVM <https://github.com/wsmoses/Tapir-LLVM.git>
Prototype implementation of the LLVM compiler with Tapir extensions for recursive fork-join parallelism.

CSI-LLVM <https://github.com/csi-llvm>
An implementation in LLVM of CSI, a framework that provides comprehensive static instrumentation.

Cilk tools <https://github.com/neboat>
A collection of dynamic-analysis tools for Cilk programs.

DotMix <https://www.cilkplus.org/download#contributions>
A deterministic parallel random-number generator for Intel® Cilk™ Plus.

PBFS <http://web.mit.edu/neboat/www/code.html>

A work-efficient parallel breadth-first search algorithm. Implementations are available for both Intel® Cilk™ Plus and Cilk++. These implementations include an implementation of the bag data structure.

Technology transfer

OpenCilk, Tapir/LLVM

Los Alamos National Laboratory developed the Kitsune parallel-aware compiler toolchain based on OpenCilk. Lucata Corporation developed a back-end to Tapir/LLVM that targets their custom in-memory-processing hardware. The design of the T4 compiler for the Swarm scalable hardware architecture is based on Tapir/LLVM. The Seq language for bioinformatics uses Tapir/LLVM to compile and optimize parallel language constructs. The TAPAS hardware-synthesis tool uses Tapir/LLVM to synthesize parallel accelerators. OpenCilk is being used for research and teaching at universities including UC Davis, Washington University in St. Louis, CMU, and MIT.

Cilk-P

Intel used Cilk-P to produce an open-source prototype library that supports on-the-fly pipeline parallelism.

Cilkprof

Intel used the Cilkprof algorithm to develop a prototype scalability profiler as a Pin tool that they now distribute.

DotMix

DotMix provided the basis for the java.util.SplittableRandom random-number generator in Java JDK8.

Pedigrees

Intel incorporated the pedigree runtime mechanism into the Intel Cilk Plus runtime and the Intel and GNU C/C++ compilers.

PBFS

Intel used PBFS to implement a parallel version of the Murphi model checker that achieves nearly perfect parallel speedup.

Technical talks

“What Compilers Can and Cannot Do”

Live-coding-demo guest lecture for 6.106: Software Performance Engineering

November 2024

“C to Assembly”

Live-coding-demo guest lecture for 6.106: Software Performance Engineering

September 2024

“OpenCilk: A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code”

“Demo: Writing Fast Task-Parallel Code Using OpenCilk”

NUWEST: NNSA-University Workshop on Exascale Simulation Technologies

January 2024

“The Cilk Runtime System”

Guest lecture for 6.106: Software Performance Engineering

November 2023

“Fast AI”

BT Insights Program

November 2023

Generative AI for Reinvention: Enabling the C-Suite

October 2023

“C to Assembly”

Live-coding-demo guest lecture for 6.106: Software Performance Engineering

September 2023

“SpeedCode: Software performance engineering education via Coding of didactic exercises”

Tutorial at SPAA

June 2023

Presented with Tim Kaler, I-Ting Angelina Lee, and Charles E. Leiserson.

“Revisiting Matrix Multiplication”

Guest lecture for 6.506: Algorithm Engineering

May 2023

"The Future of Software Performance after Moore's Law Ends"	
USGA Computing Day	April 2023
"OpenCilk: A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code"	
PPoPP	February 2023
"OpenCilk: Architecting a Task-Parallel Software Infrastructure for Modularity, Extensibility, and Performance"	
Keynote at 14th International Workshop on Programming Models and Applications for Multicores and Manycores	February 2023
"What Compilers Can and Cannot Do"	
Guest lecture for 6.106: Performance Engineering of Software Systems	November 2022
"C to Assembly"	
Live-coding-demo guest lecture for 6.106: Performance Engineering of Software Systems	September 2022
"C to Assembly"	
Guest lecture for 6.172: Performance Engineering of Software Systems	September 2021
"Panel: What's Next for Moore's Law?"	
CSAIL Alliances Annual Meeting	June 2021
"C to Assembly"	
Live-coding-demo guest lecture for 6.172: Performance Engineering of Software Systems	September 2020
"Tutorial: Research and Teaching with OpenCilk"	
SPAA <i>Presented with Dorothy Curtis, I-Ting Angelina Lee, Alexandros-Stavros Iliopoulos, and Charles E. Leiserson.</i>	July 2020
"TapirXLA: Embedding Fork-Join Parallelism into the XLA Compiler in TensorFlow using Tapir"	
HPEC	September 2019
"Tapir: Embedding Recursive Fork-Join Parallelism into LLVM's Intermediate Representation"	
Fast Code Seminar, MIT CSAIL	August 2019
"Tapir: Embedding Recursive Fork-Join Parallelism into LLVM IR"	
LLVM/Systems Seminar Series, MIT and Northeastern University	July 2019
"Ideal versus Reality: Optimal Parallelism and Offloading Support in LLVM"	
Birds of a Feather, Bay Area LLVM Developers' Meeting <i>Presented with Xinmin Tian, Hal Finkel, Johannes Doerfert, Vikram Adve</i>	October 2018
"What Compilers Can and Cannot Do"	
Guest lecture for 6.172: Performance Engineering of Software Systems	October 2018
"C to Assembly"	
Guest lecture for 6.172: Performance Engineering of Software Systems	September 2018
"Parallel Algorithms"	
Modern Algorithms Workshop, MIT CSAIL <i>Presented with Charles E. Leiserson.</i>	September 2018
"Brief Announcement: Open Cilk"	
SPAA	July 2018
"The CSI Framework for Compiler-Inserted Program Instrumentation"	
SIGMETRICS	June 2018
"Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation"	
Invited talk, University of Maryland	March 2018
Invited talk, Sandia National Laboratories	October 2017
PPoPP	February 2017
Invited talk, University of Texas at Austin	February 2017

"Principles of Tapir"		
LLVM Performance Workshop (colocated with CGO)		February 2017
"Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation"		
MIT LLVM Seminar		October 2016
"Invited Talk: Tapir: Embedding Fork-Join Parallelism into LLVM's Intermediate Representation"		
LCPC		September 2016
"Performance Engineering of Multicore Software: Developing a Science of Fast Code for the Post-Moore Era"		
Doctoral Thesis Defense		August 2016
"Deterministic Parallel Random-Number Generation, Science-Based Performance Engineering, and Life After Moore's Law"		
MIT EECS Graduating Students Day		April 2016
Invited talk, National University of Singapore		April 2016
Invited talk, Lehigh University		March 2016
Invited talk, University of Illinois Urbana Champaign		March 2016
"Three Efficient and Scalable Graph Algorithms"		
GraphDay@CSAIL		March 2016
"Analysis of multithreaded algorithms"		
Guest lecture for 6.172: Performance Engineering of Software Systems		October 2015
"The Cilkprof scalability profiler"		
SPAA		June 2015
"On-the-fly pipeline parallelism"		
Charles E. Leiserson's 60th-Birthday Symposium		November 2013
<i>Given as a joint talk with I-Ting Angelina Lee.</i>		
Invited talk, Washington University in St. Louis		October 2013
<i>Given as a joint talk with I-Ting Angelina Lee.</i>		
SPAA		July 2013
<i>Given as a joint talk with I-Ting Angelina Lee.</i>		
"Chromatic scheduling"		
Guest lecture for 6.172: Performance Engineering of Software Systems		October 2012
"Deterministic parallel random-number generation for dynamic-multithreading platforms"		
PPoPP		February 2012
MIT Industrial Liaison Program seminar talk, CSAIL series		February 2012
"A work-efficient parallel breadth-first search algorithm (or how to cope with the nondeterminism of reducer hyperobjects)"		
SPAA		June 2010
"Parallel breadth-first search using Cilk"		
Technical Seminar Series, ITA		June 2010
Invited talk, Intel Corporation		May 2010

Professional services

External service reviewer for tenure-promotion case	2024
Treasurer	2023–present
<i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	
Finance Chair	2023, 2025
<i>ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)</i>	

Associate Editor <i>ACM Transactions on Parallel Computing (TOPC)</i>	2021–2023
Program committee member <i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2019–2024
<i>ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)</i>	2022
<i>SIAM Symposium on Algorithmic Principles of Computer Systems (APoCS)</i>	2020
<i>European Symposium on Algorithms, Engineering and Applications Track (ESA — Track B)</i>	2019
<i>International Conference on Parallel Architectures and Compilation Techniques (PACT)</i>	2019
<i>ACM/IEEE Supercomputing Conference (SC), Algorithms Track</i>	2018
<i>High Performance Computing & Simulation (HPCS) Special Session on Compiler Architecture, Design and Optimization (CADO)</i>	2018
Workshop committee member <i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2020
Seminar organizer <i>Helped organize "MIT Fast Code Seminar."</i>	June 2019–present
Course facilitator <i>Organized class "CSAI-LOL: The Applications of Stand-Up Comedy" at MIT CSAIL.</i>	February–May 2019
LLVMPar coordinator <i>Coordinated LLVMPar, the LLVM working group to explore additions and modifications to LLVM's intermediate representation to support parallelism.</i>	2018–2019
Brief announcements committee member <i>ACM Principles and Practice of Parallel Programming (PPoPP) Brief Announcements Committee</i>	2019
Seminar facilitator <i>Organized the LLVM/Systems Summer Seminar series at MIT CSAIL and Northeastern University.</i>	Summer 2019
Seminar facilitator <i>Organized a seminar on LLVM at MIT CSAIL.</i>	Fall 2016
Extended review committee member <i>International Conference on Parallel Architectures and Compilation Techniques (PACT)</i>	Spring 2016
Session chair <i>Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2015, 2025
Reviewer or subreviewer <i>ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)</i>	2025
<i>ACM Journal of Experimental Algorithms (JEA)</i>	2022
<i>SIAM Conference on Applied and Computational Discrete Algorithms (ACDA)</i>	2021
<i>Elsevier Journal of Parallel and Distributed Computing (JPDC)</i>	2020
<i>ACM Transactions on Architecture and Code Optimization (TACO)</i>	2019
<i>ACM Transactions on Architecture and Code Optimization (TACO)</i>	2019
<i>ACM Computing Surveys (CSUR)</i>	2019
<i>ACM Transactions on Parallel Computing (TOPC)</i>	2018
<i>ACM Journal of Experimental Algorithms (JEA)</i>	2018
<i>ACM Journal of Experimental Algorithms (JEA)</i>	2017
<i>ACM Transactions on Algorithms (TALG)</i>	2017
<i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2017
<i>ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)</i>	2017
<i>Elsevier Parallel Computing Journal (ParCo)</i>	2017
<i>ACM Transactions on Parallel Computing (TOPC)</i>	2016
<i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2015

<i>ACM Transactions on Parallel Computing (TOPC)</i>	2014
<i>ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)</i>	2013
<i>IEEE International Parallel and Distributed Processing Symposium (IPDPS)</i>	2013

Other work experience

Principal Software Engineer (part time)	Emerald Innovations	July 2023–present
Intern	U.S. Department of Defense	Summer 2007
<i>Designed and implemented a Fuzzy ARTMap and Fuzzy ARAM in Smalltalk for the Automated Intelligence Services group.</i>		
Intern	U.S. Department of Defense	Summer 2006
<i>Developed software for the Wireless and Mobile Systems Development group.</i>		

General experience

Programming languages (in alphabetical order)

Assembly, Bash, C/C++, Cilk, Java, JavaScript, L^AT_EX, Make, Perl, Python, Scheme, Smalltalk, TypeScript, Verilog

Software technologies and systems

Compilers (LLVM, GCC), Cilk work-stealing runtime systems, Linux kernel, Intel® Pin

Relevant courses

6.856 Randomized Algorithms; 6.823 Computer System Architecture; 6.851 Advanced Data Structures; 6.854 Advanced Algorithms; 6.875 Cryptography and Cryptanalysis; 6.115 Microcomputer Project Laboratory; 6.840 Theory of Computation; 6.828 Operating Systems Engineering; 6.111 Introductory Digital Systems Laboratory; 6.035 Computer Language Engineering