eric schkufza

Berkeley CA
94709
✆ *(734) 730-4002*
✉ *eric.schkufza@gmail.com*

Research

I am interested in programming languages and compilers as they relate to high-performance computing, typically in the context of hardware accelerators. My work focuses on the optimization of machine code, often in the absence of its original source.

Education

2008–2015	**Ph.D., Computer Science**, *Stanford University*, Stanford CA. Stochastic Program Optimization for x86_64 Binaries – Advised by Prof. Alex Aiken
2006–2008	**M.S., Computer Science**, *Stanford University*, Stanford CA. Artificial Intelligence Concentration
2001–2005	**B.S.E., Computer Science**, *The University of Michigan*, Ann Arbor MI.
2001–2005	**B.S.E., Computer Engineering**, *The University of Michigan*, Ann Arbor MI.

Professional Experience

2021–Present	**Co-Founder**, *Graft, Inc.*, San Francisco CA. Building the AI of the 1% for the 99%
2020–2021	**Principal Applied Scientist**, *Amazon*, Palo Alto CA. Working on an AI-first initiative to reinvent the way Amazon does search through the use of extremely large scale next-generation deep learning techniques
2015–2019	**Senior Researcher**, *VMware Research Group*, Palo Alto CA. Bringing innovation in computer science in core areas of importance to VMware
2012–2015	**Expert Code Review**, *Feinberg Day Alberti & Thompson LLP*, Palo Alto CA. Litigating intellectual property law
2010–2011	**Research Intern**, *Microsoft Research*, Redmond WA. Improving the throughput of the Bing search engine
2005	**Junior Software Developer**, *Bloomberg, L.P.*, New York NY. Implementing search tools for financial news feeds

---
PhD Advising Experience

2019	**Joshua Landgraf**, *VMware Research Intern*. Using Cascade as a mechanism to support FPGA virtualization
2018	**Ram Srivatsa Kannan**, *VMware Research Intern*. Improving the performance of software-simulated Verilog programs
2016	**Berkeley R. Churchill**, *VMware Research Intern*. Automatically identifying third-party components in x86_64 binaries

Teaching Experience

2012   **Course Assistant**, *CS243 Program Analysis and Optimizations*.
Taught by Prof. Monica Lam, Stanford University

2011   **Course Assistant**, *CS242 Programming Languages*.
Taught by Prof. John Mitchell, Stanford University

2008   **Course Assistant**, *CS227b General Game Playing*.
Taught by Prof. Michael Genesereth, Stanford University

2005   **Teaching Assistant**, *EECS270 Introduction to Logic Design*.
Taught by Prof. Igor Markov, The University of Michigan

Professional Service

2021   **Program Committee**, *USENIX Annual Technical Conference*.
ATC 2021

2021   **Program Committee**, *Networked Systems Design and Implementation*.
NSDI 2021

2020   **Program Committee**, *Symposium on Cloud Computing*.
SoCC 2020

2018   **Program Committee**, *Virtual Execution Environments*.
VEE 2018

2018   **Program Committee**, *Symposium on Cloud Computing*.
SoCC 2018

2017   **Program Committee**, *World Wide Web Conference*.
WWW 2017

2016   **Program Committee**, *Asia Pacific Workshop on Systems*.
ApSys 2016

2008   **Co-Chair**, *AAAI General Game Playing Competition*.
AAAI 2008

2007   **Co-Chair**, *AAAI General Game Playing Competition*.
AAAI 2007

Honors and Appointments

2013–2014   **Critic**, *Architecture Studio 4*.
Department of Art and Architecture, University of San Francisco

2013   **Recipient**, *Ole Agesen Graduate Fellowship*.
VMware

2011–2012   **DJ**, *90.1 FM KZSU Stanford*.
Stanford University

2008   **Recipient**, *Computer Science Graduate Fellowship*.
Stanford University

2004–2005   **Officer**, *Eta Kappa Nu, Electrical Engineering Honor Society*.
The University of Michigan

2004–2005   **Recipient**, *Angell Hall Academic Scholarship Award*.
The University of Michigan

2001–2005   **Recipient**, *Dean's List and University Honors*.
The University of Michigan

2001–2005   **Recipient**, *Richard Earhart Academic Scholarship*.
The University of Michigan

Granted US Patents

2019   **US 10515029B2**, *Conversion Tool for Moving from Block-based Persistence to Byte-based Persistence*.
Vijay Chidambaram, Irina Calciu, Himanshu Chauhan, Eric Schkufza, Onur Mutlu, Pratap Subrahmanyam

2019   **US 20190235892A1**, *Just-in-Time Hardware for Field Programmable Gate Arrays*.
Eric Schkufza, Michael Wei

2018   **US 10162629B1**, *Compiler Independent Identification of Application Components*.
Berkeley Churchill, Eric Schkufza

2018   **US 20180095610A1**, *Methods and Systems to Sample Event Messages*.
Udi Wieder, Dahlia Malkhi, Eric Schkufza, Mayank Agarwal, Nicholas Kushmerick, Ramses Morales

2016   **US 9355470B2**, *Method and System for Interactive Layout*.
Paul Merrell, Vladlen Koltun, Eric Schkufza, Maneesh Agrawala

Case Work

2013–2014   **Code Review**, *Case No. 1:13cv740-AJT/TRJ*.
Intellectual Ventures I LLC and Intellectual Ventures II LLC, Plaintiffs, v. Capital One Financial Corporation, Capital One Bank (USA), N.A., and Capital One, N.A., Defendants

2013–2014   **Code Review**, *Case No. 13-cv-61358-rosenbaum/hunt*.
Intellectual Ventures I LLC and Intellectual Ventures II LLC, Plaintiffs, v. Motorola Mobility LLC, Defendant

2012–2013   **Code Review**, *Case No. 1:11-cv-908-SLR*.
Intellectual Ventures I LLC and Intellectual Ventures II LLC, Plaintiffs v. Motorola Mobility, Inc., Defendant

Refereed Journal Publications

Joshua Landgraf, Tiffany Yang, Will Lin, Christopher J. Rossbach, and Eric Schkufza. Compiler-driven FPGA virtualization with SYNERGY. *Commun. ACM*, 67(8):134–142, 2024.

Eric Schkufza, Rahul Sharma, and Alex Aiken. Stochastic program optimization. *Commun. ACM*, 59(2):114–122, 2016.

Refereed Conference Publications

Joshua Landgraf, Tiffany Yang, Will Lin, Christopher Rossbach, and Eric Schkufza. Compiler-driven fpga virtualization with synergy. *ASPLOS*, 2021.

Eric Schkufza, Michael Wei, and Christopher Rossbach. Just-in-time compilation for verilog — a new technique for improving the fpga programming experience. *ASPLOS*, 2019.

Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher Rossbach. Sharing, protection and compatibility for reconfigurable fabric with amorphos. *OSDI*, 2018.

Stefan Heule, Eric Schkufza, Rahul Sharma, and Alex Aiken. Stratified synthesis: Automatically learning the x86-64 instruction set. *PLDI*, 2016.

Rahul Sharma, Eric Schkufza, Berkeley R. Churchill, and Alex Aiken. Conditionally correct superoptimization. *OOPSLA*, 2015.

Eric Schkufza, Rahul Sharma, and Alex Aiken. Stochastic optimization of floating-point programs with tunable precision. *PLDI*, 2014.

Rahul Sharma, Eric Schkufza, Berkeley R. Churchill, and Alex Aiken. Data-driven equivalence checking. *OOPSLA*, 2013.

Eric Schkufza, Rahul Sharma, and Alex Aiken. Stochastic superoptimization. *ASPLOS*, 2013.

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *SIGGRAPH*, 2011.

Michael Bauer, John Clark, Eric Schkufza, and Alex Aiken. Programming the memory hierarchy revisited: supporting irregular parallelism in sequoia. *PPOPP*, 2011.

Paul Merrell, Eric Schkufza, and Vladlen Koltun. Computer-generated residential building layouts. *SIGGRAPH Asia*, 2010.

Eric Schkufza, Nathaniel Love, and Michael R. Genesereth. Propositional automata and cell automata: Representational frameworks for discrete dynamic systems. *AI*, 2008.

Refereed Workshop Publications

Tal Wagner, Eric Schkufza, and Udi Wieder. A sampling-based approach to accelerating queries in log management systems. *SPLASH*, 2016.

Himanshu Chauhan, Irina Calciu, Vijay Chidambaram, Eric Schkufza, Onur Mutlu, and Pratap Subrahmanyam. NVMOVE: helping programmers move to byte-based persistence. *INFLOW*, 2016.

Eric Schkufza and Alex Aiken. Optimizing out overcomputation. *APPROX*, 2014.

Evan Cox, Eric Schkufza, Ryan Madsen, and Michael R. Genesereth. Factoring general games using propositional automata. *GIGA*, 2009.

Eric Schkufza. Decomposition of games for efficient reasoning. *SARA*, 2007.

Technical Reports

Berkeley Churchill and Eric Schkufza. A compiler insensitive technique for identifying third-party components in x86_64 binaries. *VMware Research Group*, 2017.

Eric Schkufza. *Stochastic Program Optimization for x86_64 Binaries*. PhD thesis, Stanford University, 2015.

Eric Schkufza, Rahul Sharma, Berkeley R. Churchill, and Alex Aiken. Quantitative binary synthesis. *Stanford University*, 2014.

Eric Schkufza, Trishul Chilimbi, and James Larus. Visage: A domain-specific language for document feature extraction. *Microsoft Research*, 2011.

Nat Love, Tim Hinrichs, David Haley, Eric Schkufza, and Michael R. Genesereth. Game description language specification. *Stanford University*, 2008.