Adam Procter, Ph.D.

CONTACT

amp269@gmail.com
+1 573 819 2767
https://github.com/aprocter
http://adamprocter.com

SUMMARY

Experienced software engineer with a strong background in programming language and systems research and a proven track record of delivering solutions for highly demanding customers in the machine learning space. Interests include deep learning, compiler architecture, functional programming, semantics of programming languages, hardware synthesis from functional languages, model-driven implementation techniques for secure systems, and computer-assisted theorem proving.

EMPLOYMENT

Stripe, Inc.

Software Engineer September 2020—present

- Contributed to development of Sorbet, a open-source static type checker for the Ruby language.
- Contributed to development of the Sorbet Compiler, an ahead-of-time compiler for the Ruby language.
- Implemented compiler intrinsics and optimizations.
- Implemented tooling to triage compiler bugs as the compiler was rolled out to a larger fraction of Stripe's codebase.
- Implemented improvements to the Sorbet typechecker to improve responsiveness when used in interactive mode.
- Implemented security and reliability enhancements to the internal build and CI infrastructure for Sorbet.

Groq, Inc.

Compiler Engineer October 2019—September 2020

- Contributed to the development of front-end compiler infrastructure.
- Revamped integration with popular deep learning frameworks, resulting in a greatly improved user experience.
- Improved testing infrastructure and coverage for both front-end and back-end compiler components.
- Developed core components of the low-level Groq runtime.
- Drove roadmap discussions for the Groq software stack as part of rapid go-to-market efforts.
- Formalized and documented coding and process standards for a growing software team.

Intel Corporation

Deep Learning Software Engineer September 2017—October 2019

- Core software engineer for nGraph, an open-source compiler for deep learning models which supports many frameworks and hardware targets.
- Defined semantics of core nGraph intermediate representation.
- Designed and implemented most front-end APIs for graph construction, and many other parts of the compiler front-end.
- Made major contributions to the nGraph-TensorFlow bridge, which allows TensorFlow models to run within the framework on nGraph-accelerated backends.
- Advocated for a high standard of modularity, code quality, test-driven development, and correctness by construction.

- Represented the nGraph project at major external forums including the NeurIPS Expo and the O'Reilly AI Conference.
- Mentored numerous junior engineers contributing to the nGraph project.

Machine Learning Engineer June 2016—September 2017

- Gathered extensive, detailed, replicable benchmarking results for numerous popular deep learning workloads on pre-launch next-generation Xeon processors, which formed a major part of marketing collateral at launch.
- Developed detailed performance tuning recommendations for customers running deep learning training workloads on Intel Xeon and Xeon Phi-based clusters.
- Assisted several customers with setup and performance tuning of deep learning training on Intel Xeon and Xeon Phi-based clusters.
- Developed compelling demos of deep learning training performance for trade shows and various other public venues.
- Contributed to the development of deep learning accelerator IP for FPGAs.

University of Missouri

Postdoctoral Fellow November 2014—May 2015

- Postdoctoral researcher at the Center for High Assurance Computing.
- Developed mechanized metatheory for ReWire, a purely functional language for provably secure hardware designs.
- Research published in high-profile conferences and journals (LCTES, TECS).
- Supervisor: Professor William L. Harrison.

GAANN Graduate Fellow January 2011—November 2014
Graduate Research Assistant June 2008—December 2010

- Research assistant at the High Assurance Security Kernel (HASK) Lab, under Professor William L. Harrison.
- Developed the ReWire compiler, which translates a subset of Haskell to synthesizable implementations of provably secure hardware.

Graduate Teaching Assistant August 2005—May 2008
August 2009—December 2009
August 2010—December 2010

- Served as instructor or teaching assistant for several diferent computer science courses.
- Courses taught include Problem Solving and Programming I (CS 101), Compilers, and Principles of Programming Languages.
- Duties ranged from grading and holding office hours to teaching a large lecture course and supervising four teaching assistants.

Department of Computer Science, University of Iowa

Short-Term Scientific Employee (Summer Visitor) June 2010—August 2010

- Developed a theorem-proving system for monadic programs in collaboration with Professor Aaron Stump.

Education

University of Missouri, Columbia, Missouri USA

Ph.D., Computer Science, December 2014. Advisor: William L. Harrison.

- Dissertation: *Semantics-Driven Design and Implementation of High-Assurance Hardware*

B.A., Computer Science, *summa cum laude*, minor in Mathematics, May 2005.

PUBLICATIONS

Thomas N. Reynolds, Adam Procter, William L. Harrison, and Gerard Allwein. The Mechanized Marriage of Effects and Monads with Applications to High-assurance Hardware. *ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on MEMOCODE 2017 and Regular Papers*, volume 18, issue 1, February 2019.

Scott Cyphers, Arjun K. Bansal, Anahita Bhiwandiwalla, Jayaram Bobba, Matthew Brookhart, Avijit Chakraborty, Will Constable, Christian Convey, Leona Cook, Omar Kanawi, Robert Kimball, Jason Knight, Nikolay Korovaiko, Varun Kumar, Yixing Lao, Christopher R. Lishka, Jaikrishnan Menon, Jennifer Myers, Sandeep Aswath Narayana, Adam Procter, and Tristan J. Webb. Intel nGraph: An Intermediate Representation, Compiler, and Executor for Deep Learning. *SysML Conference 2018*, February 2018. https://arxiv.org/abs/1801.08058

Yuankun Shi, Kevin Long, Kaushik Balasubramanian, Bianny Bian, Adam Procter, and Ramesh Illikkal. DeepSim: Cluster Level Behavioral Simulation Model for Deep Learning. *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, November 2017.

Thomas N. Reynolds, Adam Procter, William L. Harrison, and Gerard Allwein. A core calculus for secure hardware: its formal semantics and proof system. *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE'17)*, September 2017.

Adam Procter, William L. Harrison, Ian Graves, Michela Becchi, and Gerard Allwein. A Principled Approach to Secure Multi-core Processor Design with ReWire. *ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on LCTES 2015, Special Issue on ACSD 2015 and Special Issue on Embedded Devise Forensics and Security*, volume 16, issue 2, April 2017.

William L. Harrison, Adam Procter, and Gerard Allwein. Model-driven design & synthesis of the SHA-256 cryptographic hash function in ReWire. *Proceedings of the 27th International Symposium on Rapid System Prototyping: Shortening the Path from Specification to Prototype (RSP'16)*, October 2016.

Ian Graves, Adam Procter, Wililam L. Harrison, Michela Becchi, and Gerard Allwein. Provably Correct Development of Reconfigurable Hardware Designs via Equational Reasoning. *Proceedings of the 2015 International Conference on Field-Programmable Technology (ICFPT'15)*, Queenstown, New Zealand, December 2015.

Adam Procter, William L. Harrison, Ian Graves, Michela Becchi, and Gerard Allwein. Semantics Driven Hardware Design, Implementation, and Verification with ReWire. *Proceedings of the 2015 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, Tools and Theories for Embedded Systems (LCTES'15)*, Portland, June 2015.

Ian Graves, Adam Procter, William L. Harrison, Michela Becchi, and Gerard Allwein. Hardware Synthesis from Functional Embedded Domain-Specific Languages: A Case Study in Regular Expression Compilation. *Proceedings of the 11th International Symposium on Applied Reconfigurable Computing (ARC'15)*, Bochum, April 2015.

Adam Procter, William L. Harrison, Ian Graves, Michela Becchi, and Gerard Allwein. Semantics-directed Machine Architecture in ReWire. *Proceedings of the 2013 International Conference on Field-Programmable Technology (ICFPT'13)*, Kyoto, December 2013.

Publications cont'd

William L. Harrison, Adam Procter, and Gerard Allwein. The Confinement Problem in the Presence of Faults. *Proceedings of the 14th International Conference on Formal Engineering Methods (ICFEM'12)*, Kyoto, November 2012.

Chris Hathhorn, Michela Becchi, William L. Harrison and Adam Procter. Formal semantics of heterogeneous CUDA-C: A modular approach with applications. *Proceedings of the 2012 Systems Software Verification Conference (SSV'12)*, Sydney, November 2012.

Adam Procter, William L. Harrison, and Aaron Stump. The Design of a Practical Theorem Prover for a Lazy Functional Language. *Proceedings of the 2012 Symposium on Trends in Functional Programming (TFP'12)*, St Andrews, UK, June 2012.

Michela Becchi, Kittisak Sajjapongse, Ian Graves, Adam Procter, Vignesh Ravi, and Srimat Chakradhar. A Virtual Memory Based Runtime to Support Multitenancy in Clusters with GPUs. *Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing (HPDC'12)*, Delft, June 2012. **(Best paper award!)**

William L. Harrison, Benjamin Schulz, Adam Procter, Andrew Lukefahr, and Gerard Allwein. Towards Semantics-directed System Design and Synthesis. Invited paper. *Proceedings of the 2011 International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'11)*, Las Vegas, July 2011.

William L. Harrison, Adam M. Procter, Jason Agron, Garrin Kimmell, and Gerard Allwein. Model-driven Engineering from Modular Monadic Semantics: Implementation Techniques Targeting Hardware and Software. *Proceedings of the IFIP Working Conference on Domain Specific Languages (DSLWC)*, Oxford, July 2009.

Pericles S. Kariotis, Adam M. Procter, and William L. Harrison. Making Monads First-class with Template Haskell. *Proceedings of the ACM SIGPLAN 2008 Haskell Symposium (Haskell '08)*, Victoria, BC, Canada, September 2008.

William L. Harrison, Gerard Allwein, Andy Gill, and Adam Procter. Asynchronous Exceptions as an Effect. *Proceedings of the Ninth International Conference on Mathematics of Program Construction (MPC'08)*, Marseille, July 2008.

Presentations

Adam Procter. nGraph: Unlocking Next-generation Performance with Deep Learning Compilers. Intel AI Developer Webinar Series, August 2019.

Adam Procter, Adam Straw, and Robert Earhart. nGraph: Unlocking next-generation performance with deep learning compilers. O'Reilly Artificial Intelligence Conference, New York, April 2019.

Adam Procter, Adam Straw, and Robert Earhart. Intel nGraph: Unlocking Next-Generation Performance with Deep Learning Compilers. NeurIPS Expo, December 2018.

Professional Service

- External reviewer for IFL'11.
- Helped organize Midwest Verification Day 2014 in Columbia, MO.

Technical Skills

Languages

- C++, C, Haskell, Standard ML, Coq, Python, Java, JavaScript, HTML/CSS, PHP, Perl, UNIX shell scripting.

Machine Learning Frameworks

- TensorFlow, Caffe, PaddlePaddle.

Methodologies
- Agile development, test-driven development.

LANGUAGES

English (native speaker)
Japanese (read, write, and speak at a high intermediate to advanced level)
German (once intermediate, now quite rusty)