Andrew Keep

Sr. ML Compiler Engineer

Redmond, WA 98052
akeep@andykeep.com
andykeep.com
akeep
andykeep

Summary

Compiler engineer with 15+ years building production compilers and language runtimes, from Chez Scheme and LLVM-based languages to ML accelerators. Currently a senior ML compiler engineer on the AWS Neuron compiler back end (allocation, scheduling, code generation) for Trainium, including Trainium3 bring-up, MXFP8/MXFP4 matrix multiply, and dynamic shapes. Ph.D. in programming languages; experienced leading small teams, mentoring, and hiring.

Skills

Languages: C++ | Python | Rust | Scheme | Scala | LLVM IR | x86_64, ARM, and PowerPC assembly

Compilers: Register and memory allocation | Instruction scheduling | Code generation | Dependence analysis | Closure conversion | Reference counting | IR design (nanopass)

ML Systems: PyTorch/JAX graph compilation | NKI kernels | MXFP8/MXFP4 quantization | Dynamic shapes | AWS Trainium

Tools: Git | Linux | macOS | Bash | Clang | Claude Code | Codex

Work Experience

Annapurna Labs, an Amazon company, *Seattle, WA* **July 2023 - Present**
Senior ML Compiler Engineer, AWS Neuron Compiler

- Technical lead for the Trainium3 compiler back end through launch, leading 3–8 compiler and kernel engineers; delivered initial Trainium3 support and the Neuron Kernel Interface (NKI) v1 implementation.
- Oversaw OCP microscaling (MXFP8/MXFP4) matrix multiply and quantization support, built by a sub-team and exposed through both NKI kernels and the PyTorch/JAX graph compiler.
- Implemented tensor indirection, enabling gather/scatter on on-chip memory (SBUF/PSUM) by making runtime-computed access patterns a general compiler feature; this became the foundation for dynamic shapes.
- Extending the back end for dynamic shapes, avoiding recompilation as batch size and context length change and enabling mixture-of-experts routing; found hardware and microcode bugs and authored ISA-check, microcode, and simulator fixes.
- Implemented loop-carried dependency-distance analysis that removes false dependencies, letting the semaphore-based scheduler overlap loop iterations across compute engines.
- Mentor engineers; conducted 90+ technical interviews for the Neuron compiler and sister teams.
- Use AI coding agents (Claude Code, Codex) daily to write and review code; built agent skills including automated regression triage that derives pass/fail criteria from a ticket, runs `git bisect`, and verifies the culprit by reverting it at head.

Meta, *Redmond, WA* **March 2020 - June 2023**
Lead Compiler Engineer (Core System Software Engineer), Reality Labs Research

- Led 5-person team implementing a new statically typed programming language targeting LLVM with a focus on AR/VR applications in the Reality Labs Research group.
- Implemented several key passes in two versions of the compiler including closure conversion, code generation, and precise reference counting.
- Implemented parts of an IDL, including a reactive values run-time for C and C++.

Cisco Systems, Inc., *Research Triangle Park, NC* **January 2014 - February 2020**
Technical Leader

- Compiler and runtime engineer on a statically typed, state-machine-based language for implementing network protocols, built in Chez Scheme with the nanopass framework.
- Co-developed a P4-14 to OpenCL compiler targeting Altera FPGAs (Chez Scheme, nanopass) and contributed to a C++ P4-16 compiler targeting the FD.io VPP software dataplane.
- Represented Cisco in P4 language standardization and at P4 workshops.
- Maintained the open-sourced Chez Scheme compiler, now the foundation of Racket and a back end for Idris.

University of Utah, *Salt Lake City, UT* **January 2013 - January 2014**
Postdoctoral Researcher

- Started development of tool for static analysis of Dalvik bytecode for Android applications, as part of a DARPA research grant.

Cadence Research Systems, *Bloomington, IN* **August 2010 - December 2011**
Software Engineer

- Rewrote the Chez Scheme compiler into passes defined using the nanopass framework.

Education

Ph.D. and M.S., Computer Science, Indiana University, Bloomington | **B.S., Computer Science**, University of Missouri, Rolla