Ian Briggs

PhD Candidate in Computer Science
(801) 574-0929 ibriggs@cs.utah.edu

PhD research scientist with expertise in numerics performance, error analysis, and optimization

EXPERIENCE

Intern, Amazon Automated Reasoning, Seattle, WA — 3 months (2022)

- Used equality saturation to rewrite and speed up queries to SMT reasoning engine
- Developed Rust simplifier using Egg library to cache and proxy queries for Z3, CVC4, CVC5
- Achieved double-digit speedups to authentication and permissions queries
- Dramatically reduced timeouts and tool instability across tens of thousands of unique queries

Research Scientist, Lawrence Livermore National Laboratory, Livermore, CA — 2 years (2018–2019)

- Developed GCC, Clang, icc, and xlC tooling to test result variability from optimization levels
- Developed root-cause and repair tool to analyze thousands of runs with different compiler flags
- Use delta debugging with Makefile/linker scripting to reduce crashes to individual functions
- Handled 16M+ lines of C/C++ code for 1000+ machine HPC clusters, found GCC bug (#90187)
- Resulting tool, FLiT, currently deployed and open-sourced

Research Scientist, University of Utah, Salt Lake City, UT — 2 years (2016–2018)

- Developed & maintained GPU symbolic execution engines GKLEE and SESA
- Tested and evaluated tool (FPDetect) to detect bit-flip errors in large (1k+ machines) clusters
- Developed LLVM fault injection tool, Vulfi, to simulate bit-flip errors

EDUCATION

Ph.D. in Computer Science, University of Utah under Pavel Panchekha — 4 years (2020–)

- Focus on numerics, low-level performance, error estimation, auto-tuning, and arithmetic
- Thesis title: "Exchanging floating-point speed and accuracy"
- 9 publications at top-tier venues: SC, POPL, HPDC, PLDI, CACM, TOPLAS
- Wrote optimizers (OpTuner, FPTuner) and DSLs (MegaLibm) for auto-tuning accuracy/speed
- Found bugs in CERN VDT library, POV-Ray caustics library; achieved speedups of 2–5×
- Also developed static (Satire, FPTaylor) and dynamic (FailAmp, FLiT) analysis tools
- Achieved best known error bounds on FFT, deployed code to HPC users

B.S./M.S. in *Computer Science*, University of Utah — 5 years (2010–2016)

- Combined undergraduate and master's CS degree, GPA 3.722
- Master's project: high-performance rigorous global optimizer using interval branch & bound
- Used in five follow-on publications at top-tier PL venues

SKILLS

Numerics: error analysis, transcendental functions, approximation algorithms, floating-point
HPC: C/C++, OpenMP/MPI, CUDA, cluster computing, Slurm, LSF, legacy code
Compilers: LLVM/LLIR, code generation, GCC/Clang, linkers, ARM & x86-64 assembly
Programming: C/C++, Rust, Python, Makefiles, CMake, Bash, SQL

PUBLICATIONS

Implementation and Synthesis of Math Library Functions — POPL
A DSL for transcendental functions that ensures mathematically correct implementations.

Choosing Mathematical Function Implementations for Speed and Accuracy — PLDI
Choosing the best in class combination of transcendental function calls for a given computation.

Keeping Science on Keel When Software Moves — CACM
Root causing result variability caused by compiler optimization.

FailAmp: Relativization Transformation for Soft Error Detection in Structured Address Generation — TACO
Detection of single event errors in address calculation in a fully automatic LLVM pass.

Scalable Yet Rigorous Floating-Point Error Analysis— SC
Improving on the state of the art to allow analysis of large floating point expressions.

Rigorous Estimation of Floating-Point Round-Off Errors with Symbolic Taylor Expansions — TOPLAS
The tightest rigorous bound for floating point error estimation.

Multi-Level Analysis of Compiler-Induced Variability and Performance Tradeoffs — HPDC
A tool to explore the effects compiler optimizations have on performance and accuracy.

FLiT: Cross-platform floating-point result-consistency tester and workload — IISWC
Examining how compiler optimizations affect computation results using a purpose built test suite.

Rigorous floating-point mixed-precision tuning. — POPL
Lowering the precision of computations to improve speed while retaining a rigorous error bound.

OTHER PUBLICATIONS

Synthesizing Mathematical Identities with E-Graphs — CoRR
FPDetect: Efficient Reasoning About Stencil Programs Using Selective Direct Evaluation. — TACO
Moving the Needle on Rigorous Floating-Point Precision Tuning. — NFM
ArcherGear: Data Race Equivalencing got Expeditious HPC Debugging — PPoPP
DiffTrace: Efficient Whole-Program Trace Analysis and Diffing for Debugging — CLUSTER