(+86)156-1611-1252
Beijing, China
jameschennerd@gmail.com

Dong Chen

HomePage: dongchen-coder.github.io
Google Scholar: Dong Chen

My research focuses on static program analysis on the correctness and performance of programs. I am interested in system software, parallel computing, programming language theories, and program synthesis.

Education

Ph.D in Computer Science, *University of Rochester* 2014.09-2019.05
BS/MS in Computer Science, *National University of Defense Technology* 2007.09-2013.12

Experience

Research Software Engineer **2022.05-**
Galois Lab, Huawei *Beijing, China*
Assistant Professor **2019.06-2021.05**
National University of Defense Technology *Changsha, China*
Intern(x2) **2016.06-2016.08, 2018.06-2018.08**
Qualcomm, Graphics Compiler Team *CA, USA*
Intern **2015.06-2015.08**
FutureWei Technologies, Compiler Team *CA, USA*

Skills

Tools and Languages C++, Python, Parallel Programming, LLVM
Communication Chinese (native), English (working proficiency)

Selected Projects

Static Analysis for Memory Safety **2022.05-**
- explores techniques to reason about program properties automatically (sparse-value flow analysis, abstract interpretation, etc).
- implements tools to identify memory bugs for large-scale industrial codes, such as null pointer dereference, memory leaks, etc.

Program Synthesis for Locality Analysis **2021.01-**
- proposes and implements an input-output-example-abased syntax-guided synthesis framework for locality analysis.
- designs a DSL and a unification search algorithm to explore the candidate program space.

Compiler Leasing **2019.01-**
- proposes a framework that enables fine-grained control of data replacements in a cache by a compiler.
- designs and implements an algorithm to derive optimal leases for each reference in a program to minimize cache misses.

Static Sampling for Locality Analysis **2018.05-2021.12**
- designs and implements an LLVM compiler pass that predicts the cache performance of loop nests. It specializes the loops to enable static profiling of reuse intervals.

Write Locality **2016.01-2016.12**
- designs and implements a linear-time algorithm to model cache writebacks from the memory access trace of a program.
- implements a scheduling algorithm to minimize writebacks by grouping co-running programs, with the writeback model.

OpenCL Performance portability **2012.01-2013.12**
- designs a source-to-source translator based on LLVM infrastructure. It automatically transforms OpenCL kernel for GPU with fine-grained parallelism to vectorized code for CPU.

Selected Publications

[Draft] *Dong Chen*. “Locality Analysis by Synthesizing Symbolic Reuse Intervals".
[TACO22] Chen Ding, *Dong Chen*, Fangzhou Liu, Benjamin Reber, Wesley Smith. “CARL: Compiler Assigned Reference Leasing".
[PPoPP20p] Fangzhou Liu, *Dong Chen*, Wesley Smith, and Chen Ding. “PLUM: static parallel program locality analysis under uniform multiplexing". 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (Poster).
[PLDI18] *Dong Chen*, Fangzhou Liu, Chen Ding, Sreepathi Pai. “Locality analysis through static parallel sampling". 39th ACM SIGPLAN Conference on Programming Language Design and Implementation. (Artifact evaluated).
[MEMSYS16] *Dong Chen*, Chencheng Ye, Chen Ding. “Write Locality and Optimization for Persistent Memory". 2nd International Symposium on Memory Systems
[HPCC13] *Dong Chen*, Changqing Xun, Dafei Huang, Mei Wen, Chunyuan Zhang. “Automatic mapping single-device OpenCL program to heterogeneous multi-device platform". 15th Conference on High-Performance Computing and Communications.

Professional Activities

Professional Services: Reviewer for JCST. Sub-reviewer for MEMSYS 2019, ICS 2019, LCPC 2018, ICS 2017, MEMSYS 2017, NPC 2017.
Teaching Assistant: Data Structure, Programming Language Design and Implementation, Advanced Compiler.