Huy Dinh Tran

huydinhtran@ku.edu
+1 (814) 826-8581
https://www.linkedin.com/in/huy-dinh-tran/
https://github.com/huydinhtran

EDUCATION

Ph.D Computer Science, *The University of Kansas* 08/2023 – present
Advisor: Dr. Mohammad Alian

M.S. Computer Engineering, *University of California, Riverside* 09/2021 – 03/2023
Project: Building custom GPU power models with AccelWattch
Advisor: Dr. Daniel Wong

B.S. Electrical Engineering, *The Pennsylvania State University* 08/2017 – 05/2021

TECHNICAL SKILLS

Languages
C/C++, Python, Bash, LaTeX, Verilog, MATLAB, Assembly Language

Tools
Visual Studio, Linux, Git, OpenMP, MPI, Gem5, GPGPU-Sim, Jupyter NoteBook, CUDA, GDB

EXPERIENCE

Graduate Research Assistant, *KU Architecture Research Group* 08/2023 – present | Lawrence, KS
- Researching on standard sampling methodologies for scale-out applications

CPU Research Intern, *Futurewei Technologies, Inc.* 08/2022 – 09/2022 | Santa Clara, CA
- Simulating CPU architecture in Linux Full-System mode using Gem5 on RISC-V ISA
- Cross-compiling binaries of SPEC CPU 2017 benchmarks to RISC-V for measuring the performance of CPU designs
- Integrated SimPoint to create checkpoints at ROIs for speeding up the simulation while still representing the workloads

PROJECTS

Building custom GPU power models with AccelWattch 03/2022 – 03/2023
- Implemented a GPU power model of an NVIDIA GeForce GTX 1050 Ti using AccelWattch
- Performed hardware profiling on real GPU to grasp the performance, real power, and hardware performance counters
- Simulated benchmarks on GPGPU-Sim to estimate constant, static, dynamic power consumptions using power model
- Achieved an average MAPE of 63.42% between simulated and real power results

Sparse matrix-vector multiplication (SpMV) 09/2022 – 10/2022
- Implemented sequential and parallel versions using OpenMP of the Sparse Matrix-Vector Multiplication algorithm in C
- Converted compressed sparse matrix formats from COO to CSR and CSC
- Achieved speedup of 3.93x between parallel and sequential computations by using 8 threads

Prefetcher using reference prediction table 09/2021 – 12/2021
- Improved a base instruction prefetcher algorithm in C++ by 5% using a reference prediction table
- Ran benchmark trace files of various workloads on a hardware simulator for measuring and comparing performance

8-Puzzle solver 03/2022 – 06/2022
- Implemented multiple tree search algorithms to solve 8-Puzzle in C++
- Implemented Uniform Cost Search, A-Star Search with Misplaced Tile Heuristic and Manhattan Distance Heuristic

Soccer matches prediction 03/2022 – 06/2022
- Implemented ML classification models in Python using scikit-learn and from scratch using NumPy and Pandas
- Implemented models: Decision Tree, Naive Bayes, K-Nearest Neighbors, Logistic Regression
- Achieved prediction accuracy of 81.25% using K-Nearest Neighbors