Siyuan Chai Contact: siyuanc3@illinois.edu https://schai.me

Education

University of Illinois, Urbana Champaign, IL
Computer Science Ph.D. Start Aug. 2021
Advisor: Prof. Tianyin Xu

Northwestern University, Evanston, IL
M.S. Computer Science, B.S. Electrical Engineering Graduated June 2021
GPA: 4.0/4.0 (Summa Cum Laude)

Publications

1. **[ASPLOS 2022]** Brian Suchy, Souradip Ghosh, Drew Kersnar, **Siyuan Chai**, Zhen Huang, Aaron Nelson, Michael Cuevas, Alex Bernat, Gaurav Chaudhary, Nikos Hardavellas, Simone Campanoni, and Peter Dinda. "CARAT CAKE: replacing paging via compiler/kernel cooperation". In Proceedings of *Architectural Support for Programming Languages and Operating Systems*.

2. **[Radiology]** Ramsey M Wehbe, Jiayue Sheng, Shinjan Dutta, **Siyuan Chai**, Amil Dravid, Semih Barutcu, Yunan Wu, Donald R. Cantrell, Nicholas Xiao, Hatice Savas, Rishi Agrawal, Nishant Parekh, Aggelos K. Katsaggelos. "DeepCOVID-XR: An Artificial Intelligence Algorithm to Detect COVID-19 on Chest Radiographs Trained and Tested on a Large U.S. Clinical Data Set." *Radiological Society of North America*.

Research Experience

UIUC Xlab, Prof. Tianyin Xu Aug. 2021 to Present
Support Linux Kernel for Elastic Cuckoo Page Table
- Adapting Linux kernel, primarily the memory management portion, to have a more versatile support for non tree page table designs like Elastic Cuckoo Page Table (ECPT), a hash page table with memory-level parallelism
- Analyzed and addressed the adaptation challenges at kernel level including page table management interface, transparent huge pages, and page table isolation
- Extensively modified address translation portion of QEMU to simulate ECPT's hardware behavior

NU Compilers Group, Prof. Simone Campanoni Jan. 2021 to July 2021
Enhance Parallelism by Utilizing Commutative Loop Iterations
- Coded a LLVM pass to transform serial code to parallel by telling the commutativity of <load, ALU operation, store> triplet across loop iterations

NU Parallelism Group, Prof. Peter Dinda June 2020 to May 2021
CARAT CAKE: Replacing Paging via Compiler/Kernel Cooperation
- Designed and implemented CARAT CAKE, an allocation level address space which aims to replace virtual memory and paging with protection checks inserted at compile time and allocations tracked in runtime
- Implemented a competitive paging address space with support for red black tree and splay tree data structures to track VA-PA mapping, transparent huge pages, and PCID; performance measured with performance monitoring counter
- Designed runtime protection check with address mapping data structures

Image & Video Processing Lab, Prof. Aggelos Katsaggelos June 2019 to July 2021
DeepCOVID-XR
- Designed and implemented a CNN model to flag out positive COVID cases based on patients' chest X-ray images
- Outperformed experienced radiologists with an accuracy of 85% compared to 76 - 82% and AUC of 0.935 compared to 0.819 - 0.856

Work Experience

Software Engineering Intern, Google Cloud Infrastructure May 2022 to Aug. 2022
Machine Model Population Pipeline

- Designed a distributed pipeline to collect data of all Google's server machines (4M+) to model their physical topology. It implements batch reads from Bigtable and capacitor or makes RPC calls with rate limitation
- Validated mac address of machines with as-maintained models across three data sources. Results will be stored in Spanner

Research Intern, Tencent Network Group June 2021 to Aug. 2021
Service Driven Network Verification tool

- Contributed to design a network verification tool for routing configurations (e,g. BGP, OSPF); it supports quantitative query and covers all data plane with global formal modeling and local simulation
- Designed easy-to-use geo-based intent language for network verification

Projects

CPU-GPU Simulator for Collaborative Workloads Modeling

- Analyzed pros and cons of state-of-art simulators (gem5, gem5-GPU, and UVMSmart) when modeling memory performance of CPU-GPU collaborative workloads
- Designed and prototyped a CPU-GPU memory subsystem simulator for workloads running on CPU-GPU unified virtual memory.
- Integrated gem5 and UVMSmart with IPC to model performance of CPU, GPU and on-demand page migration between them

Log-Structured Merge-Trees (LSM-trees) Optimization with eBPF

- Conducted a comprehensive study of optimizing LSM-trees with eBPF
- Developed two designs to offload the LSM compaction to eBPF with hook point at FS layer or NVMe driver layer

Distributed Machine Learning Inference Framework

- Developed a fault-tolerant resource-fair ML inference framework from scratch
- Coded a totally ordered KV distributed file system with quorum consensus to support the ML model inference and data storage
- Designed and implemented failure detection with a ring-based membership protocol

C-style Language Compiler, CS 322 Compiler Construction

- Created, from scratch, a compiler to translate C-style language to x86_64 assembly
- Implemented features including graph-coloring register allocation, liveness analysis, instruction selection with tiling, control flow graph, and memory access checking

Middle End Analysis for a C-based API, CS 323 Code Analysis & Transformation

- Coded a LLVM pass to reduce calls to a custom C-based API by implementing analysis like reaching-definition, constant propagation and folding, alias analysis, function inlining, and dead code elimination.

Skills

Programming languages:
C/C++, Assembly, Python, Java, Go, JavaScript, MATLAB
System-level Development:
Unix/Linux, QEMU, Docker, GDB, Make, Linker, LLVM, OpenMP
Artificial Intelligence:
CUDA, PyTorch, Tensorflow, Keras, Image Processing, Computer Vision
Hardware:
Raspberry Pi, Arduino, VHDL, Verilog
Web Development:
HTML, CSS, Flask, Django, React

Professional Activities

OSDI/ATC 2022: Artifact Evaluation Committee
SOSP 2021: Artifact Evaluation Committee, Slack Co-chair