Mahsa Rezaei Firuzkuhi

+1 (919) 985-5091 | mrezaeif@cougarnet.uh.edu | www.linkedin.com/in/mrfiruzkuhi/ | https://github.com/MahsaRF | San Jose, CA

EDUCATION

M.Sc. - Computer Science (GPA: 3.7/4), University of Houston, Houston, TX — May 2022
M.Sc. - Computer Engineering (ECE) (GPA: 3.2/4), North Carolina State University, Raleigh, NC — June 2017
M.Sc. - Computer Engineering (Computer Architecture) (GPA: 3.8/4), Sharif University of Technology, Tehran, Iran — Oct 2013
B.Sc. - Computer Engineering (GPA: 3.7/4), University of Shiraz, Shiraz, Iran — Jul 2010

TECHNICAL SKILLS

Languages	Python, C, C++, Matlab, Java, Verilog, Assembly (ARM, LC-3),
Big Data	Apache Spark Framework (PySpark), Apache PyArrow, Pandas, NumPy, fastParquet, MapReduce
Parallel Programming	MPI Programming, MPI4PY, MPI-IO, OpenMP (Familiar)
Machine Learning	Python packages such as Scikit-Learn, TensorFlow, Keras and PyTorch (Familiar), Weka
High Performance	GEM5, CACTI, Valgrind (profiling), Gprof, PAPI
NoSQL database	Apache Cassandra, Redis, Memcached

Strong Background in High Performance Computing and Low Power Architecture design (high level) with focus on Memory Subsystem Architecture, Parallel Processing for large Datasets, including Distributed-Memory Parallel Programming (MPI4PY) and In-Memory Cluster Computing for Big Data Applications (Spark framework), from system level aspects (Parallel and Distributed File Systems) and Analytics/Machine Learning algorithms for extreme scale problems

RESEARCH EXPERIENCE/ SELECTED PROJECT

Graduate Research Assistant, CS, University of Houston — Aug 2019 – May 2022

Parallel I/O operations for big-data applications using message passing interface (MPI4PY) **Git** — 2021 - Present

- Designed a high-level interface supports partitioning of file data among processes and a collective interface supporting complete transfer of global data structures between process memories and files
- Implemented the MPI4py-based read and write operations enable parallelization in big data applications through direct IO operation on most common structured file formats: CSV and Apache Parquet
- Skills used: Python, MPI4PY, Apache PySpark, Panda, Apache PyArrow, fastParquet

Parallelized calculation of hourly average of certain air pollutant in a big dataset in Spark **Git** — Fall 2020

- Developed a PySpark code to calculates hourly average of each pollutant in a big dataset of sensor-measured air pollutants applied Spark data-frames
- Leveraged NoSQL database Cassandra and compared with CSV, Parquet, and JSON file formats in terms of performance and file size as manipulating number of executors and input dataset size
- Skills used: Python, Apache PySpark, Apache Cassandra

Dog breed image classification with deep transfer learning **Git** — Fall 2019

- Utilized Inception deep convolutional neural network in TensorFlow (pre-trained Inception-ResNet-V2 and Inception-V3 model structures) to classify breed from dog image in a big dataset called "Stanford Dog Dataset" as a supervised multiclass fine-grained image classification problem
- Model a stack of a pre-trained bottleneck feature CNN with a custom output layer for application, and input images are randomly augmented to get better model generalization and achieved a validation accuracy of around 80-90%

Optimizing the Beta value of β-VAE using N-arm-bandit and Simulated Annealing **Git** — Spring 2022

- As part of a collaborative project, worked on β-VAE, a variation of the Variational Autoencoder, to improve disentangled representation learning in a complete unsupervised approach
- Proposed an n-arm-bandit and Simulated Annealing based β optimization technique, learns β value automatically based on update of loss function
- Improved image generation performance of VAE but lost reconstruction accuracy slightly

Ensemble Learning **Git** — Fall 2019

- Employed Scikit-Learn unified meta-estimators for Bagging and Adaptive Boosting (AdaBoost) to combine predictions of several base estimators built with a given learning algorithm (decision tree classifier and Support Vector Classification) to improve generalizability and robustness over a single estimator
- Determined optimal Ensemble Learning Strategy in terms of error components (bias/variance) and overfitting/underfitting in K-fold cross-validation

Cache Performance Analysis for Sort Algorithms using PAPI library **Git** — Spring 2020

- Studied behavior of different sorting algorithms by measuring data cache performance parameters, branch misprediction rate and execution time monitored by PAPI hardware performance counters
- Skills used: C, PAPI, Valgrind tool, gprof tool

Branch instruction optimization **Git** Spring 2020

- Performed branch optimization leveraging gcc macros likely/unlikely in branch condition section and PAPI library to improve branch prediction performance inside the program
- Skills used: C, PAPI, gprof tool

Machine Learning classifiers for malware code detection **Git** Fall 2017

- Designed and trained a machine Learning-based malware code detection model considering multitude of run-time hardware performance counters (HPC) as features to capture execution patterns scale a wide range of benign and malicious applications
- Utilized Linux profiler tool PERF for collecting HPC values
- Tested various type of supervised learning algorithms: Regressions, Decision trees, SVM, Random Forest bagging, Adaboost and Neural Networks
- Random Forest has shown best performance (87%-95%) in terms of True Positive rate and accuracy for all malware classes

Advanced Operating System projects **Git** Spring 2020

- Implemented a simplified shell (command language interpreter) for Linux utilizing UNIX system calls to create processes, process pipes and process communication through pipes
- Employed streaming socket to accomplish inter-process communication in a client-server connection application
- Designed thread synchronization tasks in a mutual exclusion problem with POSIX threads (pthreads), POSIX mutexes and condition variables

Graduate Student Researcher, ECE Department, North Carolina State University

Bus-Based Shared Multiprocessor Simulator to implement Cache Coherence Protocols **Git** Fall 2014

- Developed a cache coherence protocol simulator in C++ supporting MSI, MESI and Dragon coherency protocols
- Measured protocol's performance in terms of read access misses, cache to cache transfers and number of memory transactions to compare these protocols
- Architecture of Parallel Computers course project

Multi-level Cache Hierarchy Simulator **Git** Fall 2014

- Implemented a multi-level cache memory hierarchy simulator configurable in terms of inclusion and replacement policies
- Achieved a suitable cache configuration best suited for a specific application in terms of cache architecture parameters

Graduate Research Assistant, Sharif University of Technology

A Non-Volatile Processor Cache Architecture with Multi-Retention Time Intervals **Git** 2012-2014

- Designed a last level cache memory architecture using STT-RAM cells with various data retention time and write access performances, made possible by different magnetic tunneling junction (MTJ) designs
- Implemented a dynamic data redirection mechanism between low and high retention banks
- Research Assistant, High Performance Computing Architecture and Networks lab (HPCAN)
- M.Sc. Thesis: "A high-performance and power-efficient design of processor cache hierarchy using STT-RAM technology with Multi-Retention Time Intervals," M.Sc. thesis, Sharif University of Technology, 2013

Elastic Scheduling for Real-Time Tasks 2012

- Formulated a trade-off between task set schedulability and task utilization as an optimization problem enables soft real-time system to adapt to workload changes
- Implemented an optimal task compression algorithm to produce new task periods in an online manner to maintain a system-wide quality of service

Reducing Write Energy in Phase Change Memories (PCM) **Git** 2012

- As part of a collaborative project, designed a hybrid algorithm combines data manipulation and placement algorithms to reduce write access energy consumption in PCM based on minimizing number of manipulation or choosing an optimal placement for free blocks (a combination of bus inverting and content aware methods)
- Achieved %15 improvement against content aware placement technique

TEACHING EXPERIENCE

Graduate Teaching Fellow, CS Department, University of Houston 2019 - 2021

Computer Architecture, Automata and Computability

Graduate Teaching Fellow, ECE Department, North Carolina State University 2014 - 2017

Computer Systems Programming (C Programming language), Introduction to Computer Systems (Digital design and Assembly LC-3), Digital Electronics Lab,

COURSEWORK

Notable graduate-level courses:

Big Data Analytics, Advanced Machine Learning, Advanced Computer Architecture, Advanced Numerical Analysis (Convex Optimization), Machine Learning, Advanced Operating Systems, Architecture of Parallel Computers, Analysis of Algorithms, Advanced Parallel Computers, Code Generation and Optimization (Compiler), Real-Time System, Embedded Systems Design, Advanced Storage Systems, Fault-tolerant Design, Advanced VLSI design, Low Power Design, Advanced Computer Networks, Advanced Microprocessors