Keegan Smith

+1 (832) 773-8754
keeganasmith2003@gmail.com
keegancodes.com | github.com/keeganasmith

Education

Texas A&M University — College Station, TX

Master of Science — Computer Science *Jan 2025 — May 2026*
GPA: 3.875

Bachelor of Science — Computer Science *Aug 2021 — Dec 2024*
GPA: 4.0

Selected Coursework:

Quantum Algorithms	Deep Learning (PyTorch, Tensorflow, Pandas)	Software Engineering (Ruby on Rails)
Computer Architecture (C++)	Parallel Algorithms (C++, Open MPI, YGM)	Computer Networking (Visual C++)
Operating Systems (C++)	Machine Learning (PyTorch)	Programming Languages (Java, Scheme)

Work Experience

- **High Performance Computing (HPC) Software Engineer** *May 2023 - Present*
  Texas A&M University High Performance Research Computing College Station, TX
  - Benchmarked multi-core FFT and sparse matmul in C++ (**FFTW**, **oneMKL**) on Intel Optane configured as swap space
  - Helped researchers from Fermilab convert their FPGA software simulation Kubernetes workflows to Slurm scripts
  - Developed an Open OnDemand interactive application for assisting researchers with programming tasks, utilizing local llm inference to do so. Local inference was distributed across multiple nodes and performed on 4 Intel GPU Max 1100s per node. (Python)
  - Shipped an Open OnDemand interactive app to automate ML pipelines with **AutoGluon** on Slurm; standardized job specs and reduced setup time for researchers. (JavaScript / Vue.js / Python / FastAPI)
  - Automated the ticketing system by integrating with Slack using the Slack API. Created a robust workflow CLI to allow ticket workflow changes to be made without modifying code. (Python)
  - Resolved a wide range of HPC support tickets from researchers, including software troubleshooting, job scheduling issues, and environment configuration across diverse scientific domains
  - All development done on RHEL 8 machines
- **iOS Developer** *May 2023 - present*
  AKW Ventures College Station, Texas
  - Created and maintained backend for "College Football: Rapid Roster" iOS app (Swift / JavaScript / Cloudkit JS)
  - Application supports thousands of active users with an average monthly recurring revenue of $2500
  - Full Stack developer for iOS application for finding and analyzing odds across sports books. Automatically finds arbitrage, EV, and middles opportunities.
  - Application deployed using AWS lambda, AWS sqs, AWS RDS (Postgres), AWS Fargate, AWS Elastic Container Service, AWS Elastic Container Registry with Github actions for CI/CD, and Docker for containerization
  - Datafeed implemented with C++, backend webserver written in Go, frontend written in SwiftUI. Backend web server is a RESTful API
  - Backend Developer for TEC Call Auditor a tool which utilizes AI agents to grade phone calls. Utilized prompt engineering and agent orchestration to achieve 96% accuracy in production.
- **Software Engineer Intern** *May 2024 - August 2024*
  The Coca-Cola Company Atlanta, Georgia
  - Used Vue.js, Python, Flask, and Pandas to create a web application for segmenting and analyzing 8,861 outlets
  - Developed an algorithm which provided 124,054 relevant promotions for outlets

Research

- **Parallel Bridge Finding** *January 2025 - Present*
  Texas A&M University, under Dr. Roger Pearce
  - Developed and iterated upon a novel parallel bridge finding algorithm (graph theory)
  - Achieved near linear scaling up to 768 cores - 15.36 times as many cores as the current literature supports
  - Distributed computing on Lawrence Livermore National Labs systems, notably Dane
  - Implemented in C++ using the YGM library (built on OpenMPI)

Projects

- **Temporal Frequency Analysis on CPU, GPU, FPGA** *August 2024 - December 2024*
  Texas A&M University capstone project done in collaboration with Sandia National Labs
  - Optimized an existing temporal frequency analysis algorithm for CPU, GPU, and FPGA architectures
  - Won 3rd place in the Texas A&M Engineering Project Showcase Fall 2024
  - Our GPU implementation achieved the highest throughput, achieving a 125x throughput improvement over the original FPGA implementation
  - Increased throughput of FPGA implementation by 4.23x
  - Used C++, Vitis HLS, CUDA, OpenMP, JavaScript, Express.js, React.js

- **Push Battle AI** *February 2024*
  Texas A&M Datathon
  - Developed a minimax implementation for the push battle board game, and used distributed computing (MPI) to create a massive database of opening moves
  - Won 2nd place out of 600 competitors

- **M-Height Approximation** *May 2025*
  Texas A&M University CSCE 636 (Deep Learning)
  - Created a large training dataset (approximately 100 million m height problems and solutions) using OpenMPI
  - Trained a dense neural net with 1 million parameters on multiple A30 GPUs
  - Achieved a log loss of 1.20
  - Used PyTorch and OpenMPI

Technical Skills

Languages: C++, Python, Java, Bash, Swift (SwiftUI)
HPC / Systems: CUDA, OpenMP, MPI, Slurm, Linux (RHEL 8), Unix, Docker, Singularity, TCP/IP, UDP, HTTPS
Web / Tools: Flask, Vue, React, Express, Git, GitHub, GitHub Actions, CI/CD, Vim, GoogleTest
Cloud / DB: AWS (Lambda, EventBridge, Amplify, DynamoDB), MongoDB
Methodologies: OOP, Functional Programming, REST APIs, Agile, Waterfall, Test Driven Development