Bogdan Enache

enachebogdan.hahs@gmail.com | enacheb.github.io | linkedin.com/in/enacheb

EDUCATION

University of Illinois Urbana-Champaign, IL
Master of Computer Science, December 2018 GPA 4.0

University of Illinois Urbana-Champaign, IL
Bachelor of Science in Computer Science, December 2018 GPA 3.99

EXPERIENCE

Pure Storage India, Bangalore
Senior Software Engineer September 2022 - Present

- Helped establish Pure's Bangalore satellite office for the FlashBlade business unit as one of three expats, focusing on organizational and technical knowledge transfer, project management, and creating connections between the two sites.
- Provided technical guidance and oversight to a team of 9 developers working on a major feature that will generate 8 figures of revenue per year.
- Conducted extensive code and design reviews to ensure high-quality deliverables and adherence to best practices.

Pure Storage Prague, Czech Republic
Software Engineer August 2020 - September 2022

- One of three expats that helped bring up Pure's Prague satellite office for the FlashBlade business unit
- Acted as tech lead for the "Rapid Datalocking" project which shipped on time and unlocked 8 figures of revenue. Security and encryption focused, used Python. Trained the three engineers I worked with
- Acted as tech lead for the "RapidFile Toolkit" project with three other team members, increasing the performance of common Unix utilities like "find" by 16x. It was written in Go

Pure Storage Mountain View, CA
Software Engineer May 2019 – August 2020

- Part of the snapshot refactor project which decreased write latency by about 10% on snapshot heavy workloads
- Set up the pipeline to chart various metrics from customer arrays that were relevant to the snapshots team

Pure Storage Mountain View, CA
Software Engineer Intern May 2018 – August 2018

- Created a prototype application that interfaced with different enterprise products to make effiecient backups

- Worked on the project with one other intern, with guidance from a full-time manager and mentor.
- Used Python, C++, and various APIs

University of Illinois at Urbana-Champaign Champaign, IL
Teaching Assistant Janurary 2018 – Present

- Helped in the teaching of CS 450 Numerical Analysis and CS 357 Numerical Methods
- Created quizzes, homeworks, and exams
- Held office hours and discussion sessions to help students

Pure Storage Mountain View, CA
Software Engineer Intern May 2017 – August 2017

- Implemented a Docker graph driver using FlashArray as the backing store for container images
- Reduced download+run time for the Ubuntu container from five seconds to one second
- Used Go, Docker, Git, and interfaced with FlashArray

SKILLS

Best Languages: C++, Go, Python
Familiar Languages: C, Java, Rust, CUDA, OpenCL, HTML, CSS, JavaScript

PROJECTS

Senior/Masters Thesis Research Project - *Grudge*

- Grudge is a high-order, parallel Discontinuous Galerkin solver for partial differential equations (think TensorFlow for PDEs)
- Added new features like different integration schemes and support for curvilinear elements
- Implemented EM and wave equation models
- Used OpenCL and Python

COURSES

CS 473 Theory II
CS 484 Parallel Programming
CS 498 Applied Machine Learning
CS 440 Artificial Intelligence
CS 555 Numerical Methods for PDEs
CS 543 Computer Vision

COMPETITIONS WON

CS 241 Malloc

- Implemented a version of malloc using a explicit free list
- Key to victory: realizing the course staff overweighted maximum memory usage, then heavily optimizing for that (e.g. realloc would copy an entire block backwards if there was available memory), and then profiling the tested allocation patterns and optimizing for those

CS 233 Spimbot

- Wrote a bot in MIPS assembly that competed against other bots in a virtual arena
- Key to victory: teamwork and my threading implementation that allowed us to get ”energy points” with free cycles not spent on logic, and a strategy that countered other bots well

CS 296-25 Container

- Implemented an indexable data structure that supports fast arbitrary inserts and deletions, and the find-min operation (think an indexable heap)
- Used a skip list inspired data structure that held metadata in higher layers, and base layer nodes that consisted of arrays of 20 elements for cache efficiency.
- Key to victory: Good design and a lot of profiling and optimization