Raghav Prasad

rprasad@ucsd.edu • GitHub • Personal Website • LinkedIn • +1 (858) 319-9869

EDUCATION

UNIVERSITY OF CALIFORNIA, SAN DIEGO — LA JOLLA, CA
M.S. Computer Science (Systems); GPA: 3.825 — Sep 2021 - Mar 2023 (expected)
Coursework: Distributed Computing & Systems, Parallel Computing, Operating Systems, Networked Systems

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI (BITS) — GOA, INDIA
Bachelor of Engineering (Computer Science); GPA: 3.90 — Aug 2017 - Jul 2021
Coursework: Object-Oriented Programming, Data Structures and Algorithms, Computer Architecture

EXPERIENCE

COINBASE — SAN FRANCISCO, CA
Software Engineer Intern, ML Systems and Services — Jun - Sep 2022

- Engineered a custom heuristic-based approach to solve a cold start problem in the Coinbase app "For-You Feed" to ensure all pieces of content have a minimum number of impressions and potentially drive revenue
- Leveraged realtime content impressions to uprank new content without compromising metrics such as mean reciprocal rank (MRR), mean average precision (MAP), and mean area under curve (AUC) for clicks
- Conceptualized and implemented four metrics to capture the upranking effect of the heuristic and inform our choices of various hyperparameters used in the heuristic.

W.M. KECK CENTRE FOR NEUROPHYSICS, UCLA — LOS ANGELES, CA (REMOTE)
Research Intern — May 2020 - Aug 2021

- Designed and implemented the second generation of a Multimodal Virtual Reality system using C#, Unity, Python, Blender, and Arduino, increasing the system speed by up to 100% compared to previous generation
- Implemented a new computer vision based motion tracking subsystem which increased tracking accuracy by 27%
- Devised an Arduino-based multi-stream data synchronization method alongside communication protocols using websockets and serial APIs which minimized synchronization latency, improved hardware efficiency and reduced computing costs

COGNITIVE NEUROSCIENCE LAB, BITS — GOA, INDIA
Research Intern — Sep 2019 - Jul 2021

- Constructed an end-to-end PET data processing pipeline using Python, R, and shell scripts to construct and analyze brain networks to ascertain the progression of Alzheimer's Disease (AD), creating a new benchmark in PET data processing and setting a new research standard in the early detection and prevention of AD
- Engineered pipeline using parallel programming to attain up to 5-fold speedup compared to the state-of-the-art method

GNOWLEDGE LAB, HOMI BHABHA CENTRE FOR SCIENCE EDUCATION — MUMBAI, INDIA
Software Development Intern — May - Jul 2019

- Built an administrative web plugin using Ember.js, Ruby on Rails, HTML+CSS, Docker, and git. Used to curate coherent knowledge networks for an open-source learning platform resulting in an enhanced user experience for ~4000 users
- Optimized PostgreSQL database schema to enable fast search and retrieval over ~32000 articles, ensuring scalability and robustness of the plugin on larger instances of Discourse.org
- Established a template that will enable any future Discourse.org instances to be seamlessly converted into a highly scalable learning/content management system by simply integrating our plugin

PROJECTS

- Microblogging platform: Developed a scalable and fault-tolerant distributed system for users to post and view 140-character microblogs. The system can scale to 300 backends, handle concurrent requests with low-latency while maintaining causal consistency, and can tolerate up to 297 failstops without data loss
- Twitter Dapp: Developed a blockchain-based implementation of Twitter using Python and Solidity. Optimized the code to minimize latency and reduce gas fees by around 79.57% compared to a traditional distributed systems approach
- General Matrix Multiply: Co-authored a CUDA-based General Matrix Multiplication algorithm designed to run on an Nvidia GPU, modeled after CUTLASS to achieve a top performance of 621.7 GFlops/s; 60.86% the performance of cuBLAS.
- Cloud distributed file storage: Built a cloud-based distributed file storage system (like Dropbox) called *Surfstore* using Go. The system allows multiple clients to concurrently sync their local files to a server that stores files as distributed *blocks*.

SKILLS

Languages: Python • Golang • Rust • C/C++ • Java • Shell scripting • C# • React.js • HTML+CSS
Tools and Competencies: Networking • UNIX/Linux • Docker • MySQL • PostgreSQL • CI/CD • Systems programming • git