Sandipsinh Rathod

sandip@ssdd.dev | github.com/ssddOnTop | Middletown, PA 17057

Education

- Pennsylvania State University
Middletown PA
Bachelor of Science of Computer Science
Expected Graduation: Spring 2026

- **Relevant Coursework:** Formal Languages and Automata, Advanced Programming in C++

- Nirma University
Ahmedabad, GJ, India
Bachelor of Science of Computer Science
Expected Graduation: Spring 2026

- **Relevant Coursework:** Programming in C, Object Oriented Programming in Java, Linear Algebra, Data Structures and Algorithms, Introduction to Design and Analysis of Algorithms, Discrete Math, Digital Logic and Design, Computer Organization and Architecture, OS, Principles of Software Development, Introduction to Statistics
- **GPA:** 8.21

Work Experience

Tailcall

Role: Developer And open-source Contributor
December 2024 – January 2024
About Tailcall: it is the modern GraphQL runtime which helps to Swiftly design and ship best-practice GraphQL backends atop existing data sources and APIs.
Achievements/Tasks (not in any specific order):
- Abstracted code such that Tailcall can **support almost any platform**.
- Brought **custom scalars** support to Tailcall (#1214)
- Brought **gRPC reflection support** (#1647)
- Brought **Cloudflare Workers support** (and hence **WASM support**) (issue#750)
- Added (fixed) **JWT support for WASM** (#987)
- Added **Integration tests** for WASM using Miniflare (#1016)
- Added feature to **autogenerate Tailcall's config** from **Protobuf and resolve ambiguous types** (#1533, #1870)
- Brought the ability to load Tailcall config **files over HTTP** (#609)
- Added **in-memory Entity cache** for WASM (#966)
- Added TargetRuntime to specify all abstracted IOs and easily store it for Application Context
- Added proper **enum support with validation** for Tailcall config and gRPC (Protobuf) (#1537)
- Added **Jq support** (but the idea was dropped due to poor performance of Jq) (#1801, related comment)
- **Improved performance for N + 1 detection in GraphQL schema** by **Dynamic Programming** (DP) which **reduced the detection time** from **several hours** to just **a few seconds**! (#2610)
- **Introduced JIT (Just In Time) compilation** for **GraphQL** queries. (issue#2090)

Projects

Not Related to Coursework:

- Made **Augmented Reality** game of dinosaurs in 2017 (at the age of 15).
- Published first app (which was an App Lock) in 2018 on PlayStore which gained over 10k active users in just 6 months.
- In 2020, I made a **NoSQL DB** in Java. The motivation behind this project was to beat the performance of Firebase's NoSQL DB. As a result, the server was a bit faster. However it was a complete failure in terms of correctness and it was poorly structured and devoid of tests.
- I developed interest in backend services and I maintained a "homelab" using an old laptop for a few years. I heavily used KVM and Docker to ensure security and to avoid exposing actual hardware to the internet. I used some Cloudflare services as proxy atop of that.
- I hosted an Email server, personal git server, personal backup server, personal VPN which further tunnels to another VPS, etc. And all of the services were seperated as microservices in order to avoid any downtimes caused by single service. Along with this I gained a huge experience with Nginx. All the services were either served using Nginx or its reverse proxy.
- Then I added another "server" (a desktop) to different location. It was a bit more powerful then the older server, however I was facing some challenges as there was no way to get a static IP from ISP. So again I used Nginx and Cloudflare to tackle the problem and interlink both of the servers for backup.
- Later in December 2023, I participated and **won $2300 bounty** to integrate WASM for Tailcall (as mentioned above in work experience).

Related to Coursework:

- **Learning Management System (LMS)**
  - It was a group project for Principles of Software Development course.
  - The server was Developed using **Rust** and **NoSQL.**
  - LMS server can allow assignment submission, grade display, notes display and announcement notification.
  - It was built in 3 days with over 9k lines of code.
  - It achieved **100% codecoverage in** critical parts of code like authentication. With **overall 89% code coverage.**
- **Find K-mers in fasta files**
  - Under Mrs Swati Manekar, Associate professor at Nirma University, I contributed to her research by reweiting the existing C++ code to Rust and by improving performance by 3x for searching K-mers as substring in Fasta files.

Technical Skills

Programming Languages: Rust, Java, C, C++, Bash, HTML, CSS, Javascript, Matlab
other skills: Git, Github, End-to-end Encryption (E2EE), Docker, DSA, KVMs, fairly good at basic network security

Extra Curricular

Codeforces: rating of 1776, ranked top 8.5k on Codeforces

Languages: Gujarati, Hindi, English