HARSH JAIN

harshjniitr@gmail.com · github.com/harsh-98 · linkedin:@harshjain98

Skills

- Languages: **Rust**, **Go**, **C++**, **Python**, **SQL**, **Kdb+/q**
- Systems: **Linux**, **Tokio**, **eBPF**, **USDT**, **OpenTelemetry**, **perf**, **SIMD**, **memory ordering**, **async systems**
- Infra / Platform: **Kubernetes**, **Argo CD**, **Vault**, **Terraform**, **AWS**, **Linode**, **CI/CD**
- Low-level: **ARM**, **UART**, **I2C**, **memory-mapped registers**, **x86 interrupts**, **System V**

Education

Indian Institute of Technology, Roorkee July 2016 – May 2020
B.Tech. in Computer Science *8.234 CGPA*

Work Experience

Gearbox protocol 30 Aug 2021 – Present
Backend Developer(go/rust) *UAE*

- Developed a **Tokio-based Rust application** for price ingestion at **100k+ events per second**, maintaining **p99 ingestion latency below 20 ms** and **99.99% uptime** while handling price comparison, validation, and anomaly detection.
- Emitted userland **USDT probes** from the microservice using a Rust shared object extension. These **eBPF probes** were consumed by a Rust sidecar to build traces from function boundaries and request IDs, reducing resource cost versus Prometheus-based instrumentation for this workload.
- Increased observability by integrating tracing via **OpenTelemetry**, providing insights into hot paths which led to thrashing. Used **perf** to find where the extra allocations were happening on those hot paths and helped stop the thrashing of GC by using **sync.Pool**.
- Led the development of the internal **K8s cluster** and **CI/CD pipelines** from GitHub repos using **Argo CD**, with secret management being handled through the **Vault KV store** via the argocd-vault plugin. Worked on CMP server-generated plugins. Used **Terraform** for AWS resources (EC2) and for Linode node orchestration.
- While working at Gearbox, I developed an expert-level understanding of **subtyping**, **covariance**, **volatile write side effects**, **memory ordering**, **cache lines**, how to prevent optimization by the compiler, and using **SIMD**.

Squarepoint Capital LLP 10 Aug 2020 – 15 Aug 2021
Software engineer *Remote, London*

- Worked as a part of the **RISK team**; the **TDD** approach was the prime focus while writing **C++**.
- Handled deployment and bug resolution for systems, the **CI/CD pipeline** involved (**Bamboo**, **Bitbucket**, Atlassian toolset), and **Kdb+** and **q** for columnar trade data.
- Optimized **SQL queries** and **Kdb+/q analytics** on datasets with **billions+ rows**; achieved a **10x query performance improvement** through execution plan analysis.

Internship

Salesforce 5 Dec 2019 – 31 Jun 2020
Intern *Hyderabad, India*

- Conducted **performance benchmarking** of operator patterns against existing solutions (**KUDO**, **Kubebuilder**); presented **throughput** and **latency analysis**.
- Created a declarative generic tool for generating **k8s CRD operators**. Compared ease of use and scope against other inspiring projects: **KUDO**, **Kubebuilder**, and **Operatify**. The tool is available here.

Open Source

- Contributed to **go-waku**, debugged **data-race bugs**, optimized the use of **Go channels**, and reduced redundant code. Did open source for status.network for 3-4 months, handling the optimization of the **Waku Go node** for the Waku protocol at the end of 2023.
- Flashed the **micro:bit v2.2 nRF52833** using instructions written in Rust and compiled for the **ARM instruction set** to run on bare metal. Used minicom for **UART serial communication** at 112 kbps, used **I2C** for sensors, and used the PAC to read GPIO pin state directly from **memory-mapped registers**.
- Wrote an **OS from scratch in Rust**, handling **CPU/hardware interrupts**, writing to the **VGA buffer**, extending the **page table mapping** given by the bootloader, and allocating memory on the heap segments using malloc. Developed an understanding of the **System V**, **X86 Interrupt Calling Convention**. Wrote performant code for a bare-metal device with a **TDD approach**.