Siddharth Nayak

siddharth97nayak@gmail.com | siddharth1297.github.io | linkedin.com/in/siddharth1297 | github.com/siddharth1297
Noida, India

Experience

Qualcomm, Noida | *Backend Engineer* June 2024 – Present
Developing server-side components for Qualcomm's Location Technologies

- Revamping 5G positioning service to improve accuracy and multi-core throughput using Go, Redis, S3, and Datadog.
- Built a compute-intensive position measurement engine for 5G network from scratch in Go. Reduced **execution time by 23x and memory consumption by 16x** through concurrent processing and a custom memory manager.

Open Futures, New Delhi | *Software Developer* Aug. 2019 – Sep. 2021
Designed and delivered micro-second features and algorithms for in-house low-latency trading system using C++ and Python.

- **Increased profit potential by 10%** for *high-frequency automated arbitrage trading* algorithms by revamping trade execution algorithms (in C++ and Python) in collaboration with a team of 2.
- **Reduced app startup time to *1/3rd*** by porting sequential C++ code to *multithreaded* code.
- Independently, built a web-based *real-time* risk monitoring system that **slashed traders' decision-making time by 95%** using Django, WebSocket, and Redis. Wrote *asynchronous* Python HTTP and WebSocket clients for multiple crypto exchanges (**Full ownership**).

Education

Indraprastha Institute of Information Technology, Delhi Aug. 2022 – June 2024
M.Tech in Computer Science and Engineering *CGPA: 9.0/10*

Institute of Technical Education and Research, Bhubaneswar Aug. 2015 – June 2019
B.Tech in Computer Science and Engineering *CGPA: 9.3/10*

Skills

Areas of Interest: Backend Engineering, Distributed Systems, Databases, Operating Systems and Networking
Languages: C/C++, Go, Python, Java, JavaScript, SQL, Shell Scripting
Tools: Git/GitHub, Docker, Kubernetes, gdb, Valgrind, Clang-tools, eBPF
Frameworks: gRPC, LLVM, DPDK, Django, Flask, C++ QT
Databases: PostgreSQL, Redis
Cloud Platforms: AWS (EC2, S3), GCP (Compute Engine)

Projects

Serialization Performance Optimization | *Systems & Networking* May 2023 – May 2024

- Led a team of 4 to develop a serialization library leveraging *Linux scatter-gather I/O* that **outperformed Protobuf by 3x and Flatbuffers by 4x** in serialization latency within a microservices environment.
- Publication: ***Poster: Reducing Data Movement Tax for Serialization in Microservices***, in *ACM CoNEXT '24*

Fault Tolerant Distributed Key-Value Store | *Distributed Systems* March 2024

- Built a distributed key-value store from scratch using Python and gRPC, deployed over *Google Cloud Platform*. It achieved a **significantly low latency of sub-1ms for reads and sub-150ms for writes**, utilising the *Raft consensus algorithm and leader-lease technique.*

Kanva: A Lock-free Learned Search Data Structure | *Database Index* Jan. 2023 – May 2023

- Implemented a lock-free range search for Kanva (a non-blocking linearizable learned lock-free search data structure in C++), achieving **12 million operations per second on 128 cores** using a *memory-efficient constant-time snapshot algorithm.*
- Publication: ***Kanva: A Lock-free Learned Search Data Structure*** in *ICPP '24*

Argolib: A Parallel Scheduler | *Parallel Programming* Sept. 2022 – Dec. 2022

- Developed a *Fork-Join style parallel programming library and scheduler* for C/C++ programs, offering a variety of work-stealing scheduling algorithms. Additionally, **minimised runtime performance overhead up to 30%** by implementing *trace and replay* mechanism.