Siddharth Nayak

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

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 – May 2019
B.Tech in Computer Science and Engineering — *CGPA: 9.3/10*

Research Projects

Optimising Serialization for Cloud Applications [M.Tech Thesis] | *Guide: Dr. Rinku Shah* — May 2023 – May 2024

- Serialization and deserialization are two compulsory steps when remote devices communicate. In a microservice architecture, a request is processed by multiple services placed on different servers. A request undergoes (de) serialization at least once at each server, resulting in *high resource consumption* and also affects the *QOS*.
- Built a serialization library outperforming state-of-the-art libraries by 6x by leveraging Linux *scatter-gather* I/O in a microservices environment.

Kanva: Lock Free Search | *Guide: Dr. Bapi Chatterjee* — Jan. 2023 – May 2023

- Kanva is a *Learned Linearizable lock-free* search data structure with dynamic updates and range search and significantly outperforms the state-of-the-art solutions.
- My contribution: Implemented a *strong consistent(Linearizable) lock-free* range search, which offers a throughput of 12MOPS/128 cores, using a memory efficient constant-time snapshot algorithm.

Publication

Learned Lock-free Search Data Structures [preprint]
Gaurav Bhardwaj, Bapi Chatterjee, Abhinav Sharma, Sathya Peri, and **Siddharth Nayak**
To appear in 53rd International Conference on Parallel Processing – 2024 (ICPP '24)

Experience

Open Futures, New Delhi | *Software Developer* — Aug. 2019 – Sep. 2021
Designed and delivered micro-second scale features and trading 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(thread and coroutine)* Python HTTP and WebSocket clients for multiple crypto exchanges (**Full ownership**).
- Guided a junior to build an automatic log analyser platform to produce post-trade reports. Both traders and developers use the reports to analyse the behaviour of the trading strategies.

Centroxy, Bhubaneswar | *Software Engineer Intern* — June 2017 – Aug. 2017

- Developed Front-end and REST API client libraries for Python (Flask) application for Open source software Gluu.

Projects

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

- Built a **Raft** based distributed key-value store from scratch using Python and gRPC and deployed over Google Cloud Platform.
- Implemented **leader-lease technique** for reducing read latency, resulting in **significantly low latency of sub-1ms for reads and sub-100ms for writes**.

Verified Binary Search Tree in Dafny | *Guide: Dr. Piyus Kedia* Oct. 2023 – Nov. 2023

- Explored various ways of implementing a verified binary search tree using **Dafny**.
- Observed various challenges, programmer efforts, and learning curves while developing a verified program.

Study on Programmable Packet Scheduling | *Guide: Dr. Rinku Shah* Jan. 2023 – May 2023

- Programmable switches add programmability to every switch component except the traffic manager, making it only reconfigurable.
- Studied different **programmable scheduling** approaches for programmable switches and reproduced setup of SP-PIFO, a programmable scheduling technique on Intel's Tofino switch.

Argolib: A Parallel Runtime | *Guide: Dr. Vivek Kumar* Sept. 2022 – Dec. 2022

- Developed a **Fork-Join style parallel programming library and runtime** for C/C++ programs using Argobots threading library.
- Experimented *multicore scalability* of different *work-stealing* algorithms. Implemented *trace and replay* mechanisms for minimizing runtime performance overheads up to 30%. Also, implemented *dynamic concurrency throttling* for energy efficiency.

SafeC | *Guide: Dr. Piyus Kedia* Sept. 2022 – Dec. 2022

- Enhanced memory safety of C programs by writing an **LLVM** pass to catch null pointer access.
- Also implemented an automatic memory manager with a **conservative garbage collector** using the *mark-and-sweep* algorithm.

Unix Shell | *Self* Sept. 2018 – Dec. 2018

- Programmed a shell in C and implemented features like pipe, output redirection, signal handling, foreground and background processes.

Skills

Languages: C/C++, Python, Go, Java, CPython, Shell Scripting, HTML/CSS, JavaScript, JQuery, Ajax, P4, Dafny
Tools: Git/GitHub, gdb, Valgrind, clang-tools, Docker, Kubernetes, eBPF
Frameworks: gRPC, LLVM, DPDK, Django, Flask, C++ QT
Databases: PostgreSQL, Redis
Cloud Platforms: AWS, GCP

Achievements

Qualified GATE 2022
Rank-1, PGCAT-IIITD 2022

Relevant Courses

Compilers, Parallel Runtimes for Modern Processors, Concurrent and Learned Data Structures, Programmable Networking, Decision Procedures, Distributed Systems, Systems for AI, Graduate Computer Networks(seat-through)

Certifications

Machine Learning, Coursera

References

Dr. Rinku Shah, Assistant Professor, IIIT-Delhi, rinku@iiitd.ac.in *(Advisor)*
Dr. Bapi Chatterjee, Assistant Professor, IIIT-Delhi, bapi@iiitd.ac.in
Dr. Piyus Kedia, Assistant Professor, IIIT-Delhi, piyus@iiitd.ac.in