

Siddharth Nayak

siddharth22128@iiitd.ac.in | siddharth1297.github.io | linkedin.com/in/siddharth1297 | github.com/siddharth1297 | New Delhi

Education

Indraprastha Institute of Information Technology, Delhi <i>M.Tech in Computer Science and Engineering</i>	Aug. 2022 – June 2024 <i>CGPA: 9.0/10</i>
Institute of Technical Education and Research, Bhubaneswar <i>B.Tech in Computer Science and Engineering</i>	Aug. 2015 – May 2019 <i>CGPA: 9.3/10</i>

Skills

Areas of Interest: Operating Systems, Networking, Languages and Runtimes
Languages: C/C++, Go, Python, Java, CPython, Shell Scripting, JavaScript, SQL, P4, Dafny
Tools: Git/GitHub, gdb, Valgrind, clang-tools, Docker, Kubernetes, eBPF
Frameworks: LLVM, DPDK, Django, Flask, C++ QT, gRPC
Databases: MySQL, Redis
Cloud Platforms: AWS, GCP

Experience

Open Futures, New Delhi <i>Software Developer</i> <i>Designed and delivered micro-second scale features and trading algorithms for in-house low-latency trading system using C++ and Python.</i>	Aug. 2019 – Sep. 2021
<ul style="list-style-type: none">Increased profit potential by 10% for <i>high-frequency automated arbitrage trading</i> algorithms by revamping trade execution algorithms (in C++ and Python) in collaboration with a team of 2.Reduced app startup time to $1/3^{\text{rd}}$ by porting sequential C++ code to <i>multithreaded</i> code.Independently, built a web-based <i>real-time</i> risk monitoring system that slashed traders' decision-making time by 95% using Django, WebSocket, and Redis. Wrote <i>asynchronous</i> Python HTTP and WebSocket clients for multiple crypto exchanges (Full ownership).	

Projects

Serialization Performance Optimisation <i>Systems Programming</i> <ul style="list-style-type: none">Led a group of 4 to design and implement a new serialization library that reduced serialization latency by $1/6^{\text{th}}$ by leveraging <i>Linux scatter-gather I/O</i> in a microservices environment.	May 2023 – May 2024
Fault Tolerant Distributed Key-Value Store <i>Distributed Systems</i> <ul style="list-style-type: none">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 <i>Raft consensus algorithm and leader-lease technique</i>.	March 2024
Kanva: Lock Free Search <i>Concurrent Data Structures</i> <ul style="list-style-type: none">Significantly extended Kanva, a non-blocking linearizable learned lock-free search data structure written in C++, by implementing a <i>linearizable lock-free range search</i> that offers a throughput of 12MOPS/128 cores, using a memory-efficient <i>constant-time snapshot algorithm</i>.	Jan. 2023 – May 2023
Argolib: A Parallel Runtime <i>Parallel Programming</i> <ul style="list-style-type: none">Developed a <i>Fork-Join style parallel programming library and runtime</i> for C/C++ programs, offering a variety of work-stealing scheduling algorithms. Additionally, minimised runtime performance overhead up to 30% by implementing <i>trace and replay</i> mechanism.	Sept. 2022 – Dec. 2022
SafeC <i>Compilers</i> <ul style="list-style-type: none">Enhanced memory safety of C programs by writing an <i>LLVM</i> pass to catch null pointer access and an automatic memory manager with a conservative garbage collector using the <i>mark-and-sweep</i> algorithm.	Sept. 2022 – Dec. 2022

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)

Relevant Courses/Certifications

Compilers, Parallel Runtimes for Modern Processors, Concurrent and Learned Data Structures, Programmable Networking, Distributed Systems, Systems for AI