Sujal Singh

San Jose, CA | sujals9914@gmail.com |+1 (669) 250-2206 | LinkedIn | GitHub | Website

EDUCATION

Northeastern University **Anticipated June 2026**
Master of Science in Computer Science San Jose, CA

University at Buffalo, The State University of New York **June 2024**
Bachelor of Science in Computer Science Buffalo, NY
Minor in Mathematics | GPA 3.76/4.00 | Summa Cum Laude | Tau Beta Pi Honors Society | Dean's List

EXPERIENCE

Software Engineering Intern **June 2024 – October 2024**
S. S. Enterprises Remote, India

- Built and deployed a comprehensive billing and quotation management system using Node.js, React, and MySQL, reducing manual invoice processing time by 40% through automated workflows and streamlined data entry.
- Improved billing accuracy by 15% by implementing validation checks and real-time error detection, ensuring consistent and reliable invoicing across all client accounts.
- Increased customer invoice retrieval speed by 25% by integrating a user-friendly payment portal and real-time tracking dashboard, enhancing overall client satisfaction and retention.

Research Assistant (Research Paper) **April 2022 – January 2023**
Buffalo Neuroimaging Analysis Centre Buffalo, NY

- Developed an autonomous system for medical image annotation, enhancing efficiency by 40% and accuracy to 98% for over 25,000 images, reducing annotation time by 50%, and saving over 200 hours monthly, thereby improving diagnostic reliability.
- Developed a custom script leveraging Apache Spark and Hadoop to streamline DICOM data conversion, cutting processing time by 30% and improving data usability.
- Integrated XGBoost AI models on XNAT for MRI scan evaluations of neurodegenerative diseases, processing over 18,000 scans with a 20% improvement in diagnostic accuracy and a 25% reduction in evaluation time.

PROJECTS

Low-Latency Trading System Prototype (GitHub) **October 2024 – November 2024**

- Reduced end-to-end order execution latency by 35% and increased throughput by 50% by implementing a C++20-based low-latency trading system on Linux, leveraging lock-free concurrency and optimizing for CPU cache locality and Intel Xeon architectures.
- Improved TCP and multicast handling through custom network stack optimizations, achieving near real-time performance and ensuring deterministic microsecond-level latencies critical for production-critical, high-frequency trading operations.
- Enhanced system stability and scalability by integrating runtime performance engineering tools, streamlining frequent release cycles, and ensuring reliability in a fast-paced HPC environment supporting multi-threaded workloads.

FPGA Trading Accelerator (GitHub) **October 2024 – November 2024**

- Achieved sub-microsecond latency and doubled system throughput by designing an FPGA-based trading accelerator using C++ and SystemVerilog, optimizing PCIe communication and minimizing data transfer overhead on x86 hardware.
- Increased bandwidth and reduced serialization delays via lock-free algorithms, accelerating order processing on Intel Xeon servers and delivering robust production-critical systems aligned with strict performance SLAs.
- Integrated a Python interface layer atop the FPGA modules to support rapid iteration and frequent releases, enabling flexible scaling and immediate adoption of new trading strategies in a fast-paced environment.

Lock-Free Concurrent Queue (GitHub) **September 2024 – October 2024**

- Improved multi-core data throughput by 60% and reduced latency by 45% by implementing a lock-free concurrent queue in Rust, utilizing atomic operations to minimize CPU cache misses and streamline run-time behavior on Linux.
- Ensured production-grade reliability through performance engineering and asymptotic complexity analysis, enabling stable real-time data handling for high-performance, low-latency trading infrastructures.

FPGA High-Speed Networking Module (GitHub) **August 2024 – September 2024**

- Achieved data transfer rates up to 10 Gbps by developing a Verilog-based FPGA networking module, optimizing low-level hardware drivers and TCP networking stack for minimal latency and maximum bandwidth on Intel Xeon platforms.
- Enhanced system robustness and scalability in an HPC environment by fine-tuning cache-efficient data pipelines, enabling microsecond-level predictability and stable performance under production-critical loads.