SRIRAM SUBRAMANIAN SIVASAILAM

Los Angeles, California | 213-272-8481 | sriramsub7@gmail.com | LinkedIn | GitHub

EDUCATION

University of Southern California January 2025-December 2026
Master of Science in Computer Science - GPA : 4.0/4.0
Operating Systems, Distributed Systems, Hardware Foundations of Machine Learning, Analysis of Algorithms, Information Retrieval, Security Systems, Scientific Computing

PES University January 2018-December 2022
Bachelor of Technology in Computer Science
Computer Networks, Data Structures, Performance Engineering, Generic Programming, Big Data, Design Patterns, Database Systems

SKILLS

Languages: C#, Rust, C, C++, Python, SQL, Unix Shell, Regex, CUDA
Cloud: AWS, Azure, Terraform
Frameworks, Tools & Libraries: .NET Core, ASP.NET, Git, Postman, STL, NumPy, Scikit-learn, GDB, gRPC, PyTorch, TensorFlow, Claude Code

PROFESSIONAL EXPERIENCE

Software Engineer Intern May 2026-August 2026
Microsoft, WA

- Implementing optimized **distributed index** for Microsoft Defender attack signature verification to retrieve sample metadata efficiently. Optimized ingestion in service fabric and integrated with **Cosmos** Stream for uniform data access.
- Devised an **SRE Agentic AI** solution for Signature Release features under Microsoft Threat Protection Security to enable **automated** RCA for operational issues and aid on-call engineers. Integrated with **IcM**, **ADO** Code Base, and **Geneva** to enable operational support. Fine-tuned agent by refining knowledge base and providing response plans for **rule-based** decision making.

Software Engineer August 2022-August 2024
Epsilon, India

- Spearheaded migration of product to 10+ serverless computing environments, using **AWS Lambda** with **provisioned concurrency**, significantly enhancing **cost-effectiveness**
- Maintained 12+ **REST APIs** by leveraging **ASP.NET**, **.NET Core** and **C#** for database, front-end and a rules engine communication
- Upgraded automation of cloud infrastructure and resources through **Terraform** and **Azure DevOps** to manage **infrastructure as code,** improving CI efficiency
- Innovated pipelines for streamlined storage and retrieval of **PEM keys** for **AWS EC2** tenants, resulting in much **improved turnaround time** for troubleshooting of instances
- Introduced comprehensive API test suites in **ReadyAPI** and Unit Test suites in **MSTest**, exceeding expected production performance and improving code coverage by 21%
- Resolved 20+ API issues using **Postman**, ensuring seamless integration between front-end, back-end, and business logic
- Guided and trained 2 new hires and interns into the product scrum team

Machine Learning Engineer Intern June 2020-August 2020
Prodian Infotech, India

- Employed machine learning to extract most influential factors in victory/loss of domestic cricket matches in India, and worked with **Python**, **NumPy** and **Scikit-learn**
- Trained and compared **Regression**, **Gradient Boosting** and **LIME** models to cross-reference results and reduce bias by 5%

RESEARCH

Network Systems Lab USC

- Working under Prof. Seo Jin Park on research of lease management frameworks to enable rapid scale up and scale down of clusters for systems research and flash burst computing

ACADEMIC PROJECTS

GPU-Accelerated Network Intrusion Detection Engine

- Implemented an Aho-Corasick multi-pattern matching engine in **CUDA** C++ to scan network packets against attack signatures in parallel, achieving 80+ Gbps throughput on an RTX GPU
- Engineered two-phase DFA construction (trie insertion + BFS failure-link completion) enabling O(N) packet scanning regardless of signature count
- Optimized **GPU** kernel by cooperatively loading the DFA into shared memory, reducing global memory round-trips on the inner loop

Weenix Kernel From Scratch

- Implemented processes, threads, virtual files and virtual memory for an operating system kernel based on 6th edition **UNIX** using **C**
- Wrote **bootstrap** functionality, created the idle process and thread context to support process and thread management
- Created a virtual file system that supports compatible actual file systems through **polymorphism** and implements reference counting for efficient virtual node and file descriptor management
- Implemented address space management to support user level processes with efficient process forking using **shadow objects** and **shadow walking** to facilitate Copy-on-Write

Kite-Shell
- Architected a high-performance Unix shell with a dynamic **System-Aware Prompt** in **C++17** using **POSIX** system calls, implementing a complete REPL lifecycle with custom tokenization and process management
- Integrated an AI-assistive subsystem using **llama.cpp**, enabling natural language-to-command translation and automated error correction by injecting shell-state context into LLM prompts

LLM Inference Optimization (Llama 3.2-1B)
- Developed a high-performance inference codebase for Llama 3.2-1B, implementing INT4 weight quantization and Mixture-of-Experts (MoE) to optimize model footprint and latency on A40 GPUs
- Implemented **LoRA** (Low-Rank Adaptation) and slice modes for MoE, evaluating trade-offs between perplexity, next-token accuracy, and throughput (tokens/sec) using **PyTorch** and **CUDA**

Raft Consensus for Distributed State Machine Replication
- Implemented leader election and Raft consensus log replication to support self-reliant distributed systems using **gRPC** and C++
- Integrated a **RIFL** layer to enable exactly-once semantics for strong consistency, efficiency, safety and liveliness

Automated Tool for C Source Optimization
- Programmed an automated tool for **optimization** of **ANSI C** code using **Python Lex-Yacc**, compared correctness and utilized meta-programming to encode 3 metric calculations into programs for performance benchmarking
- Integrated 14 techniques based on Jon Bentley's rules and general coding paradigms which supplement compiler optimizations and work at code level

Yet Another Centralized Scheduler
- Conceived a centralized scheduling framework for jobs in a distributed system and enabled the integration of 3+ **Big Data** scheduling algorithms and can be specialized based on use case to boost throughput by 30%+
- Implemented a socket-based communication protocol to manage real-time resource heartbeats and dynamic task allocation across a cluster of distributed worker nodes

Generic Splay Tree
- Created a **generic splay tree** container in C++ which supports 5 operations to utilize for **caching and optimized memory** of user-defined data, achieving up to 90% hit in web caches
- Implemented a bidirectional iterator to enable the generic splay tree to be seamlessly plugged in to existing STL algorithms

PUBLICATIONS

Teaching Aid For Program Optimization

K. M. Meda, S. Subramanian, A. Bennur, S. Vijay and N. S. Kumar, "Teaching Aid For Program Optimization," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021