Shrikanth Sudhersan

shrikanthsudhersan10@gmail.com linkedin.com/in/ssudher github.com/ssudher 919-345-5968

EDUCATION

North Carolina State University, Raleigh, NC, USA August 2019-May 2021
Masters in Computer Science, GPA : 4/4
Software Engineering, Operating System, Parallel Systems, Game-Engine Foundations, Algorithms, Internet Protocols

Amrita Vishwa Vidyapeetham, India August 2012-May 2016
Bachelor of Technology in Electronics and Communication Engineering, Distinction

TECHNICAL SKILLS

Language/Database: Python, C, C++, C#, OpenMP, OpenMPI, OpenAcc, CUDA C, SQL, MongoDB
Tools/Frameworks: JIRA, GitHub, GitLab, Wireshark, GDB, Hadoop, Spark, MapReduce, ZeroMQ

WORK EXPERIENCE

Synopsys, North Carolina, USA May 2020-August 2020
Software Engineering Intern

- Worked on enhancing the existing product line by integrating a C++ based proprietary distributed programming library
- Designed programs to stress-test the library in a distributed environment to identify potential failure scenarios
- Built compute-intensive services to emulate workload of the product to gauge performance and resilience of library
- Redesigned the application to improve cache reuse and cache-hit ratio. Boosted the performance by 20%
- Collected data points, identified best practices, prepared documentation for fault-tolerant usage of distributed library

Honeywell, Bangalore, India January 2016-April 2019
Senior Software Engineer

- As a part of the R&D team, used Agile and Scrumban techniques to conceptualize a unified framework using Python and C# for automating the collaboration of development team and testing team. This reduced wastage by 40%
- Programmed a method using Python to parse the source code(C/C++) to build a custom code comparison module
- Used LEAN methods in development of a chat-assistant that understands user intents and gives useful insights into industry jargon. Leveraged Python's (NLTK) Natural Language Tool-Kit, C# and SQL to devise the framework
- Employed TDD concepts to automate the development of specific types of test-cases in Python for several categories of requirements that eliminated manual effort and human prone mistakes. This reduced the cycle time by 30%
- Recognized as the 'Outstanding Achiever' of the year 2017

PROJECTS

Game Engine | **C++** August 2020-December 2020

- Designed and implemented a modular game engine from scratch following SDLC processes. Used Inheritance, object-oriented design concepts, multi-threaded game loop architecture and tested the software using BDD methodology
- Incorporated multi-player support using ZeroMQ with custom serialization & de-serialization methods from scratch

XINU Operating System | **C** August 2019-December 2019

- Implemented memory virtualization through demand paging using swapping algorithms: Second Chance and Aging
- Designed a robust scheduler(exponential-distribution, linux-like) and lock(reader/writer) with priority-inheritance

Point-to-Multipoint File Transfer | **Python, Socket programming** August 2019-December 2019

- Using the underlying principles of UDP as the transport layer protocol implemented a multi-threaded program that uses ARQ methods to achieve reliable file transfer from one sender and multiple receivers

Bit-Torrent | **Python, Socket programming** August 2019-December 2019

- Implemented a peer to peer file sharing model using REST APIs to allow distributed peers to exchange files

Cache Coherency Simulator | **C++** August 2019-December 2019

- Implemented bus based cache-coherency protocol MSI,MESI,Dragon for a shared L1 cache in a multi-core processor
- Simulated and analyzed the protocols for their performance by varying cache size, block size and associativity

Simplified Custom Message Passing Directives | **C, OpenMPI, pthreads** January 2020-May 2020

- Implemented MPI directives such as Sendrecv, Barrier, Init, Gather and Finalize using producer-consumer pthread functions and socket programming in C

Accelerating Graph algorithm | **C, OpenMP** August 2019-December 2019

- Designed a memory-efficient adjacency lists for building Compressed Sparse Row (CSR) using Radix sort