Sharadh Rajaraman

+44 7512 070176 | +65 9846 7310 | r.sharadh@outlook.sg | sharadhr | sharadhrajaraman

Summary

Software engineer at PetaGene Ltd, a subsidiary of Storj Inc; working on Object Mount, which provides POSIX-compatible file access to object storage, with a focus on performance and reliability; targeted at media, healthcare, and high-performance computing (HPC) customers.

Enjoys supporting the tech community through open-source contributions. Have a deep passion for looking under the bonnet of software and hardware, and approaching problems with a scientific mindset. Effective communicator and team player, with a strong interest in mentoring and teaching.

Singapore citizen, with UK Skilled Worker visa. Eligible for High Potential Individual (HPI) and H-1B1 visa programs. Open to opportunities in the UK, Europe, and Singapore.

Work Experience

PetaGene Ltd

Cambridge, United Kingdom

Software Engineer (C++, Objective-C, macOS and Windows desktop programming, CMake, Jenkins, C#/Avalonia, Windows driver programming)

Aug 2023 – Present

- Spearheaded build tools transition from GNU Autotools and shell script to vcpkg and CMake, thus reducing build and CI run times by a factor of 10, enhancing developer productivity and enabling IDE usage.
- Introduced optimisations for thumbnails presented by Object Mount in Windows Explorer and macOS Finder. Improved responsiveness for media and entertainment customers by reducing network traffic by 90%. Used by a major film studio.
- Implemented optimisations across network mounts on macOS, allowing for server-side copy and deduplication.
- Applied static analysis and code quality tools on the codebase, allowing for early detection of bugs and security vulnerabilities, improving code quality, and reducing technical debt.
- Represented the company at SuperComputing 2023, the largest HPC conference in the world in Denver, CO. Attracted ~100 leads, and several new customers.

National University of Singapore (NUS) School of Computing (SoC)

Singapore

Undergraduate Teaching Assistant

Aug 2020 – Nov 2022

- Taught classes in computer graphics, real-time rendering, introductory programming, and computer architecture.
- Set up auto-grading harness for computer graphics assignments to automate marking by comparing framebuffers and pixel errors.
- Conducted weekly tutorials and recitations, prepared materials and videos for students, and marked assignments.

Government Technology Agency, Singapore (GovTech)

Singapore

Embedded Software Engineering Intern (Sensors and IoT Division: C/C++, CMake, STM32)

May 2022 – Aug 2022

- Implemented a C++ wrapper over Linux Serial Peripheral Interface (SPI) syscall interface. Reduced wheel-reinvention, and improved linkage for other projects using C++.
- Reused COVID-19 contact-tracing tokens using STM32 microcontrollers, by rewriting firmware in C++ to emulate a Trusted Platform Module (TPM) over I^2C for Raspberry Pi (rPi). Reduced costs by recycling and reusing existing hardware.

Skills

Languages C++, Objective-C, C#, Python, Java, LaTeX, GLSL/HLSL, F#, TypeScript, PowerShell

Frameworks & Tools AWS SDK, Foundation, Win32, OpenGL, DirectX, Vulkan, OpenMP, CUDA, IDA Pro

Projects

Personal projects and contributions

vulkan.cppm (Merge request)

C++20 module for Vulkan-Hpp (C++, CMake)

- Adapted a code generator to output a C++20 module interface file for the Vulkan-Hpp wrapper library.
- Improved type safety and performance by exporting C macros and function-like macros as `constexpr` variables and functions.

Coursework

Oat Compiler

Compiler for statically-typed, C-like Oat language with Python-like list comprehension (OCaml, Menhir)

- Front-end outputs a subset of LLVM IR; back-end lowers to a subset of x86_64 assembly.
- Static single assignment (SSA) design to enable optimisations.
- Implements a type system with type inference, expression and statement typing, and type covariance/contravariance.
- Compile-time optimisations e.g. constant and expression folding, dead-code elimination, and register allocation with graph colouring.

Static Program Analyser

Lexer and parser for a C-like toy language (C++17)

- Lexer implemented with `std::regex` state machine; straightforward greedy algorithm.
- Recursive-descent LL parser, with left-recursion elimination and operator precedence parsing.
- Inserts information such as variable declarations, function calls, and control flow into a database about a given program written in the toy language.

Also enjoyed implementing:

cache-sim (C++20, CMake), which implements MESI, MOESI, Dragon cache-coherence protocols for a quad-core CPU.

Education

National University of Singapore *Singapore*

Bachelor of Computing (Honours) in Computer Science; 2nd Major in Physics *Aug 2018 – May 2023*

- Computer science: parallel computing; real-time computer graphics; operating systems; compiler design.
- Physics: astrophysics; quantum mechanics; solid-state physics.

Extracurriculars

Indian classical music

Carnatic vocal, violin *Sept 2003 – Present*

- Completed diploma in Carnatic vocal music in 2012 and violin in 2013
- Performed solo since 2010 in Singapore, India, Australia, and the UK
- Conducted workshops and classes on Carnatic music theory and practice for beginners