Puyan Lotfi

puyan at puyan dot org
github.com/plotfi

Georgia Institute of Technology Atlanta, GA
MS Computer Science May 2011
BS Computer Science May 2007

Meta: Compiler Engineer (PL&R, Oculus) Menlo Park, CA (May 2018 - Present)

- **AIML/GPU Compiler Engineer:** Early member of a new team focused on GPU and ML compiler development around the Triton DSL. Have been exploring the tradeoffs between L1 caching and use of shared memory.
- **Swift C++-Interop for Facebook Apps (PL&R):** Drove an effort that involved Meta engineers collaborating with the Swift compiler community's C++-Interoperability effort. The value-add for developers was greater productivity by avoiding the need of manually writing as many C wrappers around C++ headers for use from Swift. I worked cross functionally with orgs who build UI libraries as well as cross company (**https://bit.ly/3zfY3YF**) with the open source community to achieve progress on this project. Much of my involvement was in experimenting and using real app code from framework developers to drive an agenda of making this feature of the Swift compiler more usable. I was part of a cross-company Swift compiler working group on these matters. Much of my involvement was in ironing out bugs and issues related to C++-Interop is the presence of the existing Objective-C interop (**https://bit.ly/3hJhimH**). This effort culminated in the release of Swift 5.9 and Xcode 15.
- **iOS App Size Reduction (PL&R):** Wrote a size compression transformation that involved targeting part of the language's metadata in order to reduce app size (2MB/slice) and also reduce app startup time due to smaller metadata needed to load at launch.
- **Clang+LLVM Release Engineering (PL&R):** I was instrumental in providing insights on how to do better engineering in my team's LLVM codebase by either refactoring private commits into more easily git rebase-able components, upstreaming code, understanding past technical debt and knowing when it can be deleted, and also pushing for better culture around testing and code hygiene. My contributions helped to make it possible to release an up to date toolchains much more frequently. My greatest achievement on this project was in affecting team culture and teaching my team better practices.
- **Clang Interface Stubs + llvm-ifs (Oculus):** Clang Interface Stubs is a feature in upstream clang and llvm whose utility is in providing library authors a way to directly generate stub-libraries from clang ASTs where said stubs only contain symbols explicitly marked as visible. It was conceived as a way to bring tooling similar to Microsoft Import Libraries and Apple's TAPI to ELF. I gave a **2019 LLVM Dev Meeting Lightning Talk (https://bit.ly/35ap4uk**) on this project. Originally the goal was for use by an Oculus AR/VR Operating system (the OS was put on hold). However as it turns out my work was adopted and expanded by Google's Fuchsia OS team (**https://fuchsia.googlesource.com/fuchsia/+/5e786c3**); I collaborated with engineers on Fuchsia in reviewing patches so that their adoption would go more smoothly.

Apple: Backend GPU Compiler Engineer (LLVM-GPU Team) Cupertino, CA (August 2013 - May 2018)

- **GPU Compiler Backend:** I have worked on a variety of backend codegen components of Apple's LLVM GPU compiler for iOS in various capacities including the Fast Instruction Selector (effort lead by me for iPhone 6 and 6s), code scheduler (bug fixes), compile time tuning, as well as adding new features and encodings for new hardware Targets. I have experience working through multiple product life cycles that include Apple A8, A9, A10, A11, and A12 based devices as well as iOS 8-12 OS releases. I did compiler backend work on Apple's first ever in-house GPU design (The A11 Bionic GPU).
- **llvm-mc based GPU Assembler:** I lead the effort to build Apple's first ever internal LLVM based GPU assembler (A11/A12 GPU) while also doing the first ever assembly syntax design as well. I used a BNF format for writing a syntax description document that was used cross-functionally as part of an assembly syntax unification effort of all assembly for GPU tools across all of Apple.
- **GPU Binary Tooling Infrastructure and Disassembler:** Wrote a disassembler in pure C++ to verify instruction encodings coming out of the LLVM based compiler and assembler with the goal of catching encoding bugs extremely early in the development process of future hardware. Said disassembler ended up turning into a general binary tooling infrastructure with its own IR and APIs. It was used at Apple for everything from building shader profiling tools to doing performance studies. For performance studies this tooling infrastructure was used to build a GPU simulator where compiled shader assembly was transformed into the x86 machine code in order to simulate future GPU hardware that did not exist at the time.
- **Metal Shader Profiler:** As part of a cross-functional tooling effort aimed at building a publicly available shader profiling tool, I wrote a library based on the above mentioned binary tooling infrastructure that provides program counter information of places in code where shader pipeline execution gets interrupted. This binary level shader analysis library was used by my counterparts to build the Metal Shader Profiler. Aside from pipeline profiling, mentioned GPU Binary Infrastructure was used to make cost breakdowns for instruction types in a given shader. The Metal Shader Profiler shipped in Xcode 10. **Apple Developer Documentation Link:** https://apple.co/2Loj9NW. **US Patent 10310830:** https://patents.google.com/patent/US10310830B2
- **MIR-Canon: Improving Code Diff Through Canonical Transformation.:** Implemented a machine pass for canonicalizing Machine IR for cleaner textual comparison. Presented this work **at the 2018 Euro LLVM conference:** https://bit.ly/39P3ikG. **Video link:** https://youtu.be/RHT-bh_xo6U
- **Metal Explorer:** Worked with GPU Driver org to deploy a modified Godbolt's Compiler Explorer for use with Metal shaders and assembly. This project has been and continues to be a huge success with many internal users across various hardware and software orgs at Apple (even to this day: metalexplorer.apple.com)

Intel: GPU Compiler Engineer (C for Metal/Media, aka "CM") Santa Clara, CA (June 2011 - July 2013)

- Backend compiler engineer architecture and the original author of CM's GPU printf: **https://bit.ly/3xttLz5**; also authored CM's bytecode

ArrayFire: Intern Compiler Engineer Atlanta, GA (Summer 2010)

- Wrote and shipped a Flex+Bison parser that was used for writing element-wise Matlab GPGPU kernel lambdas