JEFF NIU

Email: jeffniu22@gmail.com
Mobile: +1 650 864 3841
Github: github.com/mogball
LinkedIn: in/jeffniu22

work experience

MODULAR *2022-now*
- Lead architect for Mojo, a Pythonic systems programming language for high performance heterogenous computing, with first class Python and C++ FFI
- Gave a bunch of talks about this work
- Built Mojo's MLIR-based implementation from scratch: LLVM code generation, mid-level optimizer (inliner, SROA, mem2reg, arg promotion, AA, memory SSA, etc.), metaprograming features, compile time interpreter, type system, compilation model
- Implemented tons of major language features across the stack: async/await, traits and generics, closures, type families/constraints, parameter inference, metatypes, exception/error handling, split GPU compilation, lifetimes and memory safety
- Prototyped key Modular technologies in Mojo that were handed to other teams: NDBuffer with partially static properties, thread pool and async runtime, structural codegen prototype, "engine extensibility API"
- Lead integration of Mojo and Modular's graph compiler, tuned the mid-level optimizer for autofusion, and drove engineering excellence across both teams
- Drove >1000x reductions in compile time over 2 years (Mojo compiles llama3 with hundreds of specialized kernels for SoTA performance in less than 40 sec)
- Prototyped and co-designed tile-based programming model in Mojo that outperforms Nvidia vendor libraries (CUBLAS, CUTLASS, etc.) with 100x less engineering effort
- Technical Lead of the Mojo Compiler Team, driving language design and evolution with 6 other engineers across the frontend and backend, primary design and technical PoC with all other teams at Modular, DRI for many projects (debugger, Python interop, GPU bringup)
- Daily balancing of the long-term vision of the language against short-term business needs
- Performed various managerial/TPM functions: "therapy" and growth mentorship for engineers, quarterly work planning, prioritization, and allocation, bug triage, talent sourcing and hiring, etc.
- Hosted an intern to build first-class library optimizations, paper targeting PLDI 2025

**GOOGLE (*CoreML)* *2021-2022*
- Built high-level control-flow and schedule optimizer for TensorFlow, delivering ~2% improvements to Google AdBrain models ($millions of compute/year)
- Implemented compile time optimizations to the TensorFlow compiler, saving ~$700k/yr
- Built MLIR dataflow analysis framework, collaborated with JSIR team to implement security analyses over JavaScript code running in Chrome and other Google environments
- Contributed to upstream MLIR core infrastructure, dialects and tooling

internships

**ETH ZURICH (*Scalable Parallel Computing Lab)* *2020*
- Built IRDL, a DSL for defining compilers, published in PLDI 2022

CITADEL *Summer 2020*
- Bringup and evaluation of Cerebras hardware for internal HPC applications

GOOGLE *(Brain)* *Fall 2019*
- Built PDL, an optimized pattern rewriter, now used by various hardware companies

APPLE *(Silicon validation)* *Winter 2019*
- GPU assembly programming to stress SoCs as much as possible

COREAVI *(Embedded graphics)* *Summer 2018*
- Hacking on embedded GPU driver for various Khronos standards, VxWorks 6/7, etc.

YAHOO! *(Big data)* *Fall 2017*
- Contributed to Apache Superset, built anomaly detector, created ember-localforage

projects

gg-mlir: An MLIR-based distributed workload optimizer and executor (slides)
WATERLOOP: C++ STL and package manager, embedded networking and architecture
UWNRG: Computer vision, pathfinding, robotics stuff

education

UNIVERSITY OF WATERLOO: B.A.Sc. in Mechatronics Engineering (2021)