Robert VanDerzee

✉ *robbie@vanderzee.co*
robbie.vanderzee.co

Skills

Languages	C++23, C, Java, Rust, MLIR, Python, JavaScript, OCaml, Assembly, SQL
Toolchains	LaTeX, Mathematica, MATLAB, Vulkan, LLVM, Abseil, GTest, React, Google Sanitizers
Systems	Kubernetes, AWS, Bazel, Protobuf, Spring, WebAssembly, MongoDB, Cassandra, Redis
Domains	Compilers, Operating Systems, Graphics, Parallelism, Networking, Distributed Systems

Experience

2024–Present **Compiler Engineer (MAIA)**, ***Microsoft***, C++, Python, Triton, MLIR, LLVM.

- Designed and implemented Triton compilation discovery mechanisms, saving hundreds of engineering hours by enabling parallel pre-compilation and linking.
- Developed a distributed Triton cache, minimizing compilation invocations with a 10% team-wide reuse rate; improving kernel development throughput by approximately 15%.
- Created a LIT testing framework for our Python DSL, deprecating the hand-written LIT test process.
- Implemented FX-Graph and TTIR fusion pattern matching, improving MAI inference performance by 20%.
- Bound MLIR runtime errors to Python DSL, enabling immediate error feedback for enhanced user experience.
- Interviewed and subsequently trained new hires, supporting a rapidly growing team of 50+ engineers.

2021–2024 **Machine Learning Infrastructure Engineer**, ***Apple***, Java, Scala, Rust, C++, Bazel, AWS.

- Built a task scheduler and database abstraction service with parallel request execution from dependency trees.
- Extended a multivariate experimentation platform with stochastic schedule-based allocations.
- Developed a data pipeline query service for visualization and experimental analysis.
- Supported more than 100 million low-latency requests per day with service optimizations.
- Implemented a caching layer for database query aggregation, increasing throughput by 50%.

Projects

2023–Present **Molten**, Bazel, Rust, Molten.

Event-based nondeterministic finite-state machine graph compiler written in Rust.

- Architected a mathematically pure graph-based intermediate representation for abstract compute.
- Designed and implemented a text-based front-end targeting the graph-based IR.
- Created a matrix-vector based graph runtime enabling parallel execution.
- Built a visually rendered, non-textual programming interface.
- Testing a NDFSM optimizer using reinforcement learning to overcome an incomputable search space.

2019–Present **Pure Mathematics**, C++, Mathematica.

Various studies of pure mathematical domains, namely Number Theory, visualized with *Mathematica* tools.

- Atypical prime number (-1, 2, 3, 5, ...) translation, factorization, extension, and decomposition.
- Complex analysis and complex representation analogues and extensions.
- Arithmetic progressions such as Bernoulli numbers and the Hailstone sequences.
- Pascal sequences (1; 1, 1; 1, 2, 1; ...) and Pascal matrices, recursive variation analyses.

Education

2018–2021 **Bachelor of Science**, ***The University of Virginia***, School of Engineering and Applied Science.

Majors: Computer Science, Physics

Courses: Algorithms; Artificial Intelligence; Calculus; Circuit Design; Classical Mechanics; Compilers; Computer Architecture; Computer Graphics; Data Structures; Discrete Mathematics; Game Design; Linear Algebra; Operating Systems; Ordinary Differential Equations; Probability; Software Engineering; Theory of Computation

Thesis: *P-computation: Algorithmic analyses for prime representations of higher-dimensional numbers*

Achievements

2022–2023	**E-sports (Codeforces)**, Expert (Rating: 1633), Top 13%.
2019–Present	**E-sports (Chess)**, Rank: 45,792 / 3.29M, Top 1%.
2016–2019	**E-sports (League of Legends)**, Grandmaster (Rank: 1,000 / 1.7M, NA), Top 0.06%.
2019-Present	**Language**, English (Native), Korean (Professional), Chinese (Beginner).