Marisa Kirisame

Email | Website | GitHub

Education

University of Utah, PhD in CS *2020 –*
University of Washington, Master in CS *2019 – 2020*
University of Washington, Bachelor in CS *2015 – 2019*

Projects

Spineless Traversal, Lead Developer

Implemented an incremental computation algorithm that beats SOTA web layout by 3.23×.

- Novel incremental attribute grammar algorithm that reduces cache misses
- Staged compilation DSL; compiler in OCaml, output high-performance C++
- Complex data structure (order maintenance, priority queue)
- Custom allocator, hand-written assembly, cmov to reduce branch prediction, rdtsc for measurement

Kirisame et al. "Spineless Traversal for Layout Invalidation", submitted to PLDI 2024, [2411.10659].

MemBalancer, Lead Developer

Reduced garbage collection time by 30% for V8, the JavaScript engine behind Chrome.

- Built and solved mathematical model of garbage collection for optimal heap size
- Control theory to smooth input signal to maintain stable heap size
- Implement for V8; 4k lines of code, concurrency, runtime metrics, integration with allocator, task system
- Patch accepted to V8, collaborations leading to patches in Firefox, Racket, MMTk, Guile Scheme

Kirisame, et al. "Optimal Heap Limits for Reducing Browser Memory Use." In *OOPSLA*, 2022

DTR, Lead Developer

Enabled PyTorch Neural Networks to be trained using as little as 20% of the original memory budget.

- Recast an autodiff problem into a runtime system problem, emitting plans better than classical technique
- Deep modifications into PyTorch internals, spanning dozens files, sophisticated interaction with autodiff
- Work with engineers at Megengine and Oneflow to get it adopted in the industry

Kirisame, et al. "Dynamic Tensor Rematerialization." In *ICLR*, 2021

TVM, Compiler Developer

Contributed to the design of Relay, a higher order, differentiable IR.

- Top 20 contributor to TVM, a high performance ML compiler-runtime
- Implemented Algebraic Data Types, Automatic Differentiation(AD), Ahead-Of-Time Compiler
- Used Partial Evaluator, and Continuation Passing Style Transform to reduce AD overhead

Skills

Expertise: Incremental Computing | Compiler | Garbage Collection | Staging & Partial Evaluation | Program Analysis | Algorithm & Data Structure | Functional Programming | Automatic Differentiation | Proof Assistant | SMT Solver | Computer Architecture
Familiarity: Systems Programming | Make & CMake | Git | Operating System | ML System | Database
Programming Languages(Proficient): C++ | Coq | Haskell | OCaml | Python
Programming Languages(Familiar): Assembly | C | Scala | Verilog
Languages(Proficient): English | Mandarin | Cantonese