

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

Implemented an incremental web layout program that beat the SOTA by $3.23\times$.

Wrote a compiler in OCaml that converts attribute grammar to an incremental program, using a novel incremental attribute grammar evaluation algorithm which accesses much fewer nodes than the SOTA. Implemented a highly efficient order maintenance data structure, which is used by the algorithm. Use a custom hand-written allocator to improve cache locality, and inline assembly with conditional move to reduce branch misprediction. Use the compiler to convert a layout algorithm into an incremental program which is then evaluated against the SOTA.

Marisa Kirisame, et al. “Spineless Traversal for Layout Invalidation,” 2024

MemBalancer

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

Derived an optimal formula for heap sizing with Calculus and Lagrange Multipliers. Using control theory, implemented a controller that measure garbage collection/allocation speed, smoothed the two signal, and set the heap limit to the optimal value from the formula. Utilize concurrent programming and garbage collection knowledge.

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

DTR

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

Developed an algorithm for gradient checkpointing for large machine learning model, by evicting tensors to save memory, and recursively recomputing them when needed later. Adopted by Megengine and DELTA. Choose the stalest, largest, cheapest to recompute tensor to evict, and avoid evicting long chain, so the total time overhead is a small fraction even though memory saving is large.

Marisa Kiris, et al, me. “Dynamic Tensor Rematerialization.” In *ICLR Year = 2021*, n.d.

TVM

Top 20 contributor to high performance ML compiler-runtime. Contributed to the design of Relay, a higher order, differentiable IR. Implemented Algebraic Data Types, Automatic Differentiation, Ahead-Of-Time Compiler, Partial Evaluator, and Continuation Passing Style Transform.

SKILLS

Expertise: Incremental Computing | Compiler | Garbage Collection | Program Analysis | ML System | Proof Assistant | Functional Programming | SMT Solver | Computer Architecture | Operating System

Programming Languages: Assembly | C | C++ | Coq | Haskell | OCaml | Python | Scala | Verilog

Languages: English | Mandarin | Cantonese