

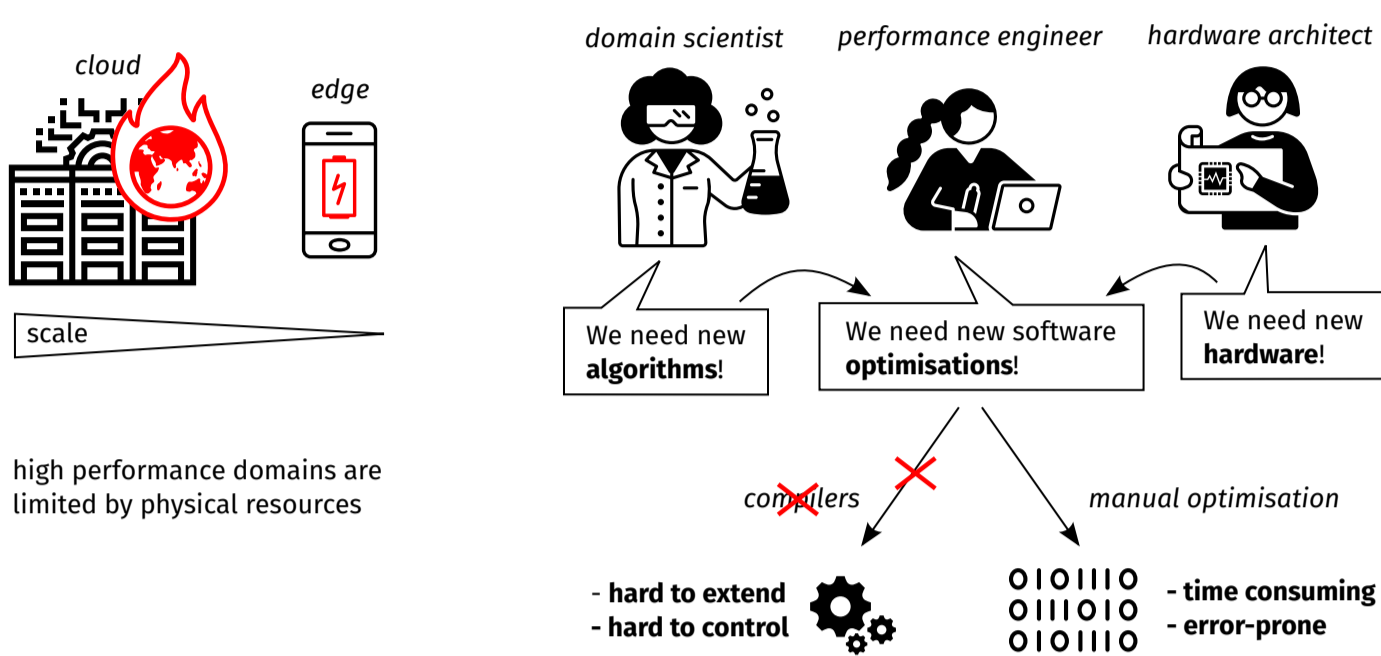
A Domain-Extensible Compiler with Controllable Automation of Optimisations

Thomas Kœhler
Phil Trinder, Michel Steuwer



PhD Viva, December 2022

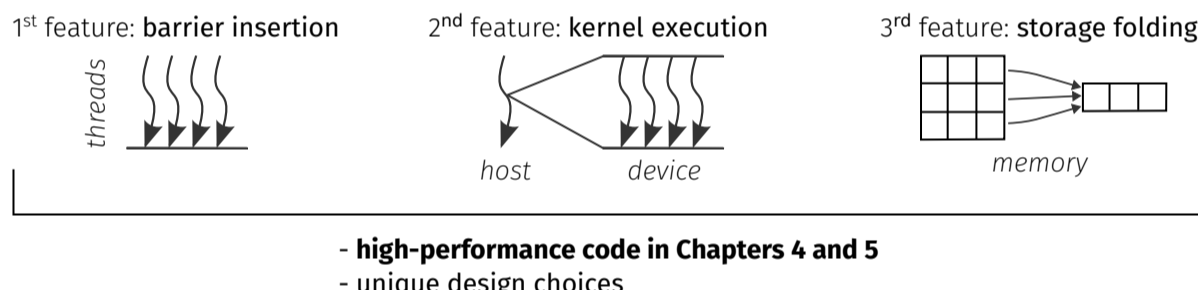
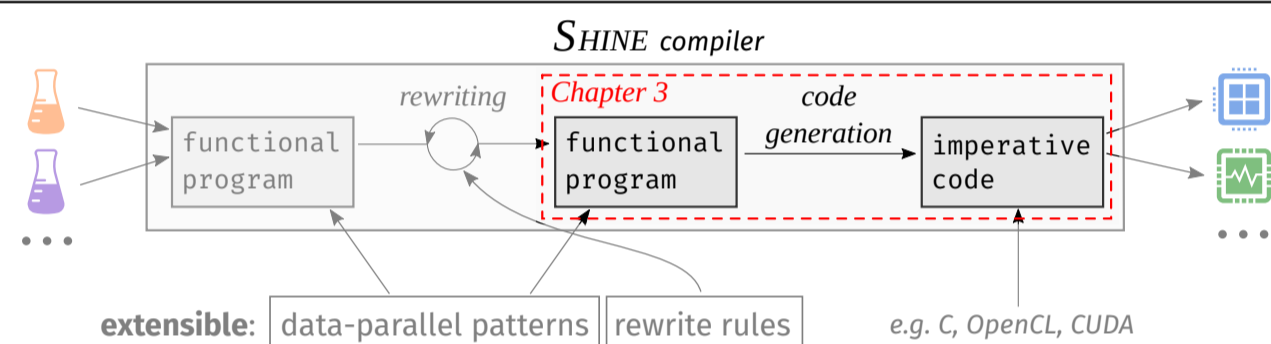
Problem: Performance Engineers Bypass Compilers



Thesis explores how effectively the SHINE compiler achieves:

- **domain-extensibility**: extensible abstractions and optimisations
- **controllable automation**: smooth automation/control trade-offs
- **high performance** code generation

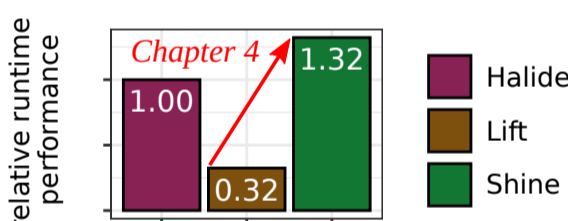
1st Research Contribution: Enhancing Code Generation in a Domain-Extensible Compiler



2nd Research Contribution: Going Beyond Halide Scheduling with Controlled Rewriting



Case Study:
- standard corner detection pipeline
- 6 well-known optimisations
- 4 ARM multi-core CPUs



domain-extensibility + controlled rewriting
=> additional optimisations
=> **generates faster code**

First Author of **CGO'21 paper [1]**
Co-author of **ICFP'20 paper [3]** (CACM Research Highlight)

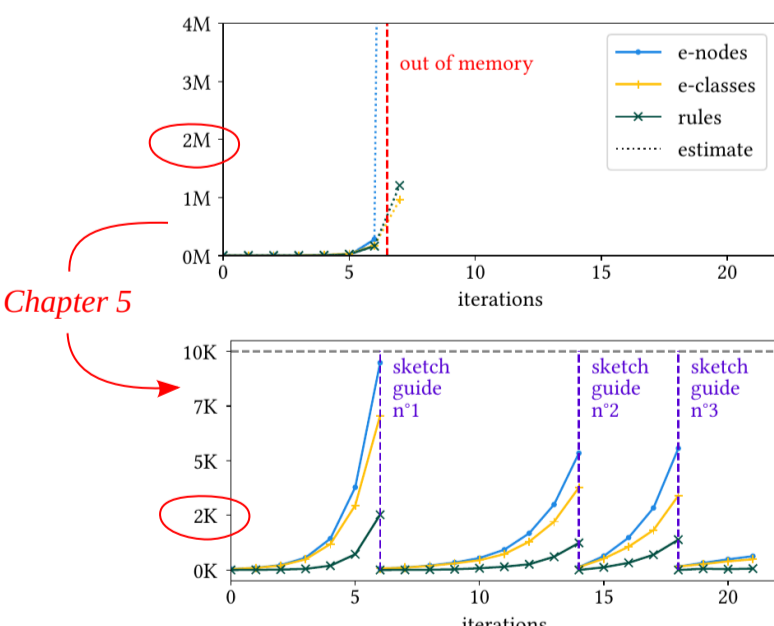
3rd Research Contribution: Proposing a Novel Semi-Automatic Optimisation Technique

Sketch-Guided Equality Saturation:
guide rewriting by specifying goals as sketches

Case Study:
- 7 optimisations of matrix multiplication

unfeasible unguided search (1h, 60GB RAM)
+ sketch-guidance (≤ 3 relatively small guides)
=> **feasible search** (<10s, <1 GB RAM)

First Author of **arXiv paper [2]**
Invited Talk at **EGRAPHS Workshop, PLDI'22**



Vision: Performance Engineers Cooperate with Compilers

