A shared compilation stack for distributed-memory parallelism in stencil DSLs

George Bisbas 1  Anton Lydike 2  Emilien Bauer 2  Nick Brown 3  Mathieu Fehr 2  Lawrence Mitchell 2  Gabriel Rodriguez-Canal 2  Maurice Jamieson 3  Paul H. J. Kelly 1  Michel Steuwer 4  Tobias Grosse 5
1 Imperial College London, UK  2 University of Edinburgh, UK  3 EPCC, University of Edinburgh, UK  4 Technische Universität Berlin, Germany  5 University of Cambridge, UK

authors equally contributed

The problem: Monolithic Domain-specific languages

Tailored to their domain, but actually lots of common generic concepts!
✓ Performance✓ Productivity✓ Portability

Technical challenges:
- Independent/Siloed
- Lack of code reuse
- Separate
- Short lifespan

Societal challenges:
- Disjoint communities
- Lack of knowledge transfer

We propose: Using compiler technology!

Contributing a shared compilation stack for HPC in stencil DSLs:
✓ Performance✓ Productivity✓ Portability

Technical benefits:
- Composability
- Code reuse
- Interoperability
- Longevity

Societal benefits:
- Connected communities
- Extensive knowledge transfer

Our work enables reuse of HPC and target-specific abstractions across DSL and compiler frameworks and offers synergies across DSL communities while maintaining the community-tailored interfaces of each DSL compiler.

Proposed solution

- Idea: MLIR offers a unified IR but needs bridging with HPC concepts.
- Fill the gap: Introducing HPC-specific abstractions for interoperability with MLIR dialects.
- How: Utilizing xDSL, a python-native clone of MLIR, and building HPC abstractions.
- Case: Focus on explicit finite difference (FD) stencil computations as a representative case study.

Contributions

- An SSA dialect to facilitate automated domain decomposition for distributed-memory execution of stencil kernels via message-passing.
- An SSA dialect for message passing as a set of modular operations in a standardized SSA-based IR.
- Upstream to MLIR!
- A prototype implementation of abstraction-sharing compilation stack for two HPC stencil-DSL compilers, PSyclone and Devito, based on the concepts of SSA and Region and utilizing the MLIR and xDSL compiler frameworks.
- A performance evaluation demonstrating that our approach is competitive for a range of FD stencil computations, compared to the existing domain-specific compiler stacks, for CPU shared- and distributed-memory parallelism, GPUs and FPGAs running at scale on ARCHER2 and Cirrus.

References