A common compiler ecosystem for domain specific languages

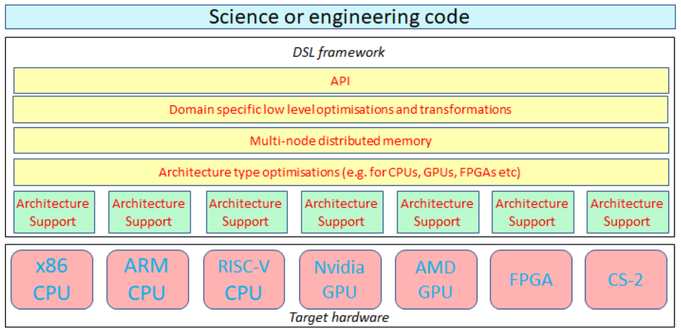
xDSL A common compiler ecosystem for domain specific languages

Corresponding Author: Nick Brown (EPCC, n.brown@epcc.ed.ac.uk)

epcc | Imperial College London | The University of Edinburgh informatics

What are Domain Specific Languages (DSLs)?

There is an explosion of hardware architectures for HPC, and as we move into the exascale era a key challenge is how to fully exploit such complex, highly parallel heterogeneous supercomputers.



DSLs provide a separation of concerns and are arguably the only way in which we can effectively program and exploit future exascale supercomputers

Why are DSLs not ubiquitous in HPC?

A major disadvantage is in the siloing of compiler infrastructure, where DSLs tend to share very little or no infrastructure between them at the toolchain level.

This siloing of toolchains results in:

- ✗ Uncertainty around long term maintenance
- ✗ Reinventing the wheel
- ✗ Considerable effort needed to develop a DSL
- ✗ Limited opportunities for DSLs to target multiple domains
- ✗ Limited third-party tools
- ✗ Considerable effort for new architectures

We must solve the challenges around siloing in order for DSLs to become more widespread

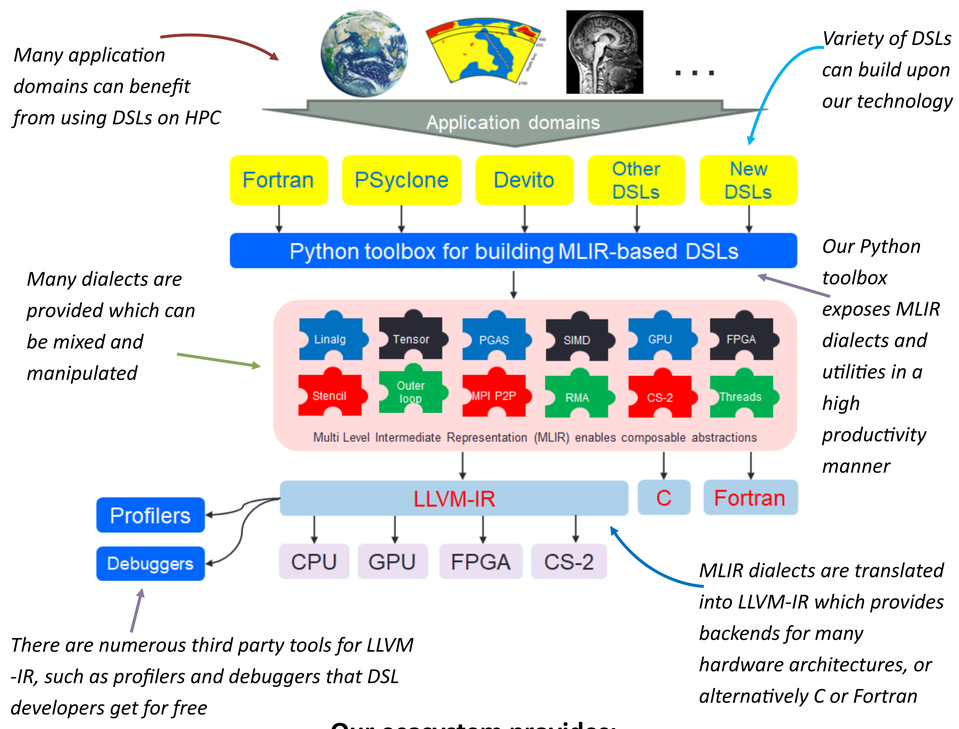
Hence our vision: A DSL being a thin layer atop an existing, mature ecosystem with a wealth of third party tools

30 second summary

- Domain Specific Languages (DSLs) have great potential.
- Underlying toolchains are often siloed and share very little or no infrastructure.
- We are developing a Python ecosystem so DSL developers can write a thin-layer on-top of the existing, well supported, MLIR/LLVM compiler infrastructure.

What is our xDSL ecosystem?

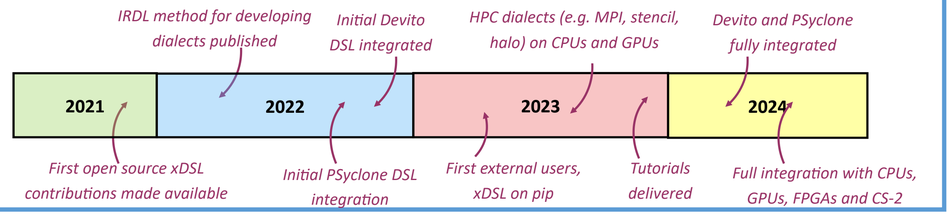
A Python toolbox based upon MLIR, integration with MLIR & series of HPC dialects



Our ecosystem provides:

- ✓ ***Composability*** where DSL owners choose what parts of our ecosystem to leverage
- ✓ ***Interoperability*** between DSLs
- ✓ ***Code reuse*** of toolchain infrastructure
- ✓ ***Longevity*** of DSL compiler technology as we build upon LLVM and MLIR
- ✓ ***Performance*** because dialects and backends are developed by experts
- ✓ ***Productivity*** for both the application and DSL developers due to code reuse
- ✓ ***Portability*** for application codes and DSLs across architectures.

Project running August 2021 until July 2024



Building on LLVM and MLIR

LLVM is a collection of common compilation tools and infrastructure. There are numerous LLVM backends available for different hardware, targeted via LLVM-IR.

MLIR

MLIR enables representing and mixing dialects of intermediate representations and abstractions, thus providing easier integration, reuse, and optimisation. However, currently written in C++ there is a fairly steep learning curve that our Python toolbox looks to address.

Get Involved

xDSL is open source and a burgeoning community

www.xdsl.dev
github.com/xdslproject
xdsl.zulipchat.com

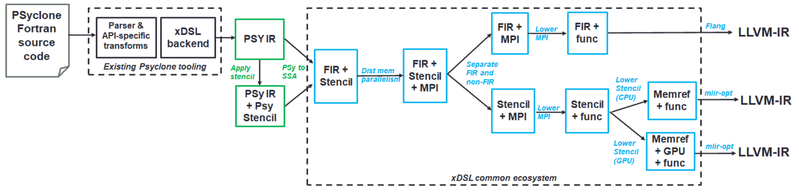
Evaluating the xDSL framework

We have selected two existing DSLs to act as evaluators

Devito is a Python based DSL for symbolic finite difference computation. Especially popular for seismological simulations, there are a variety of users from oil & gas exploration to glaciologists

PSyclone is a Fortran based DSL for developing weather and climate codes. It is used extensively by the Met Office

PSyclone



Above illustrates translating the PSyclone DSL's Fortran code into LLVM-IR using our approach

Funded by ExCALIBUR

ExCALIBUR

https://excalibur.ac.uk/

The UK ExCALIBUR program address the challenges and opportunities offered by computing at the exascale and aims to deliver the next generation of HPC simulation software and tooling