Legacy code base on HPC accelerators

Source-to-source translation for column based numerical algorithms

Balthasar Reuter, Michael Lange, Olivier Marsden

Research Department, ECMWF, Reading, United Kingdom

The objective

Target a broad range of HPC architectures including **accelerators** (such as GPUs or FPGAs) and achieve **good computational performance** from a **single code base**.

The challenge

- Different **programming paradigms** and environments
- **Hardware-specific optimization** (loop order, block size, ...)
- **Large** and **complex code base** needs to be adapted

The code base

Developed for **physical parameterizations** ("physics") of the Integrated Forecasting System (IFS) but concept applicable for larger class of **column based numerical algorithms**.

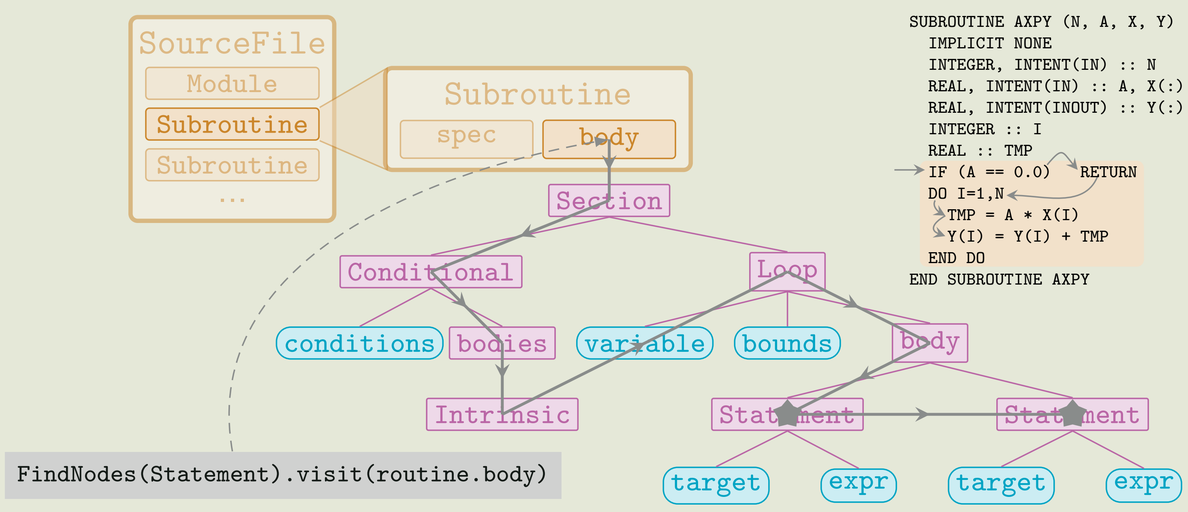
- Physics model subgrid-scale processes in the atmosphere
- **No data dependencies** between columns
- Expectation: **up to 38% of operational cost** in the future[1]

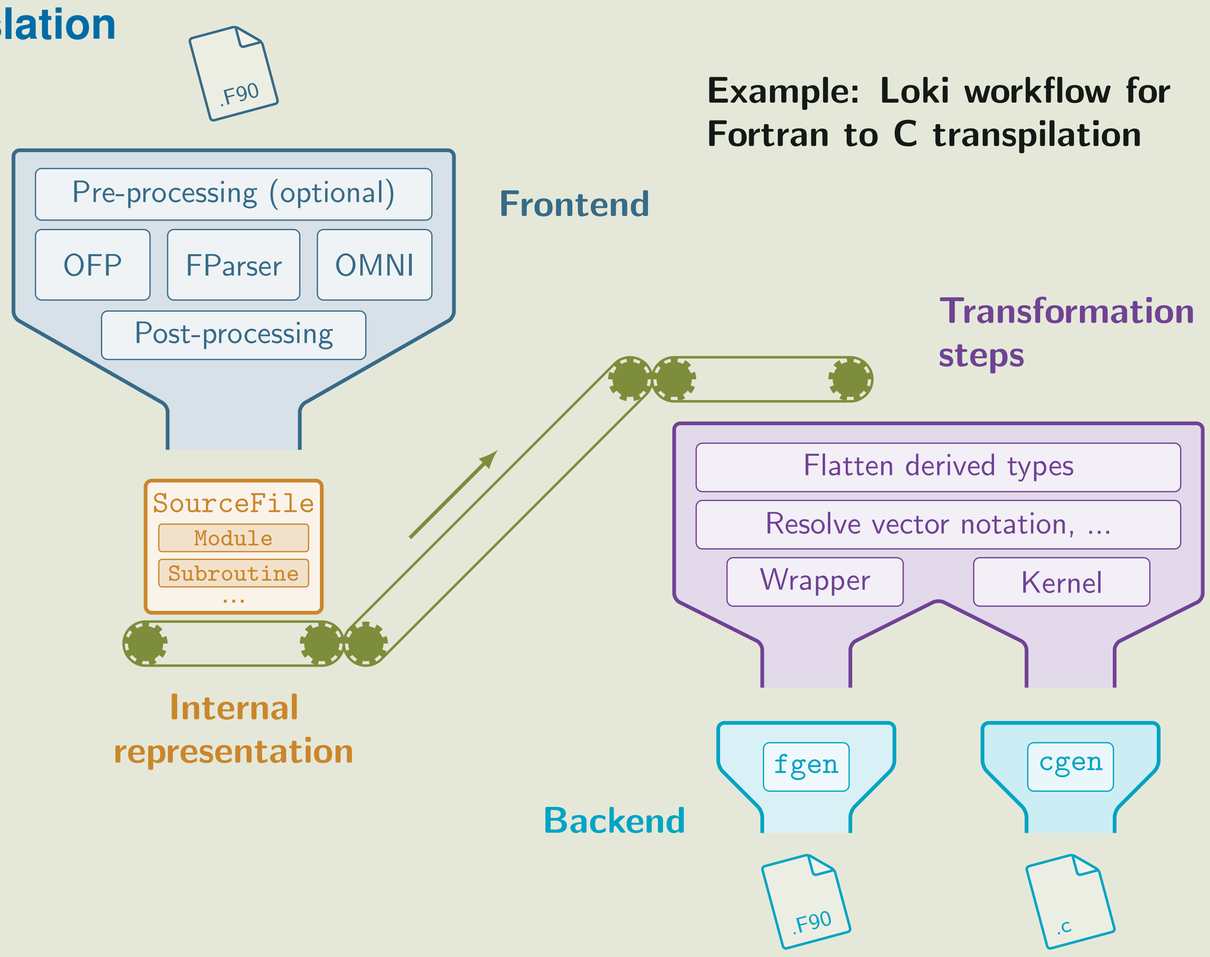
The approach

- Apply **source-to-source translation** to apply bespoke transformations to Fortran source code
- **Feed into downstream tools** (such as CLAW[2] or MaxCompiler) to produce hardware-specific versions
- Exploit **single column abstraction** to reorder horizontal and vertical loops as required for target architecture

Loki: a tool for user-guided automated source-to-source translation

- **Purpose-built** in Python for the IFS
- Offers an API to **encode custom transformations**
- Bespoke **internal representation (IR)** based on trees
- **Two-level IR**: Expression trees based on *Pymbolic*[3]; custom data structures for statements / control flow
- **Traverse** and **transform IR** using **visitors**

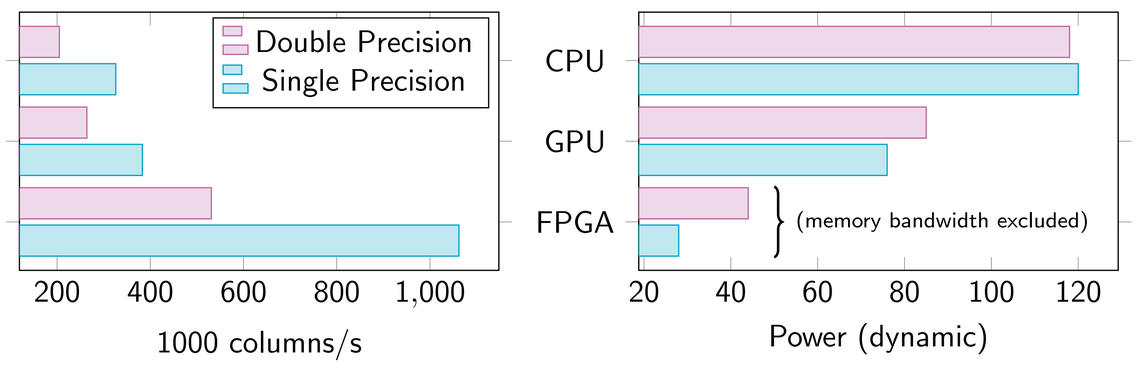




- **Frontend** employs *Open Fortran Parser*[4], *FParser*[5], or the *OMNI Compiler*[6] frontend to read Fortran source files
- **Multiple backends** to generate Fortran, C, or MaxJ code

Use case: CLOUDSC on GPU and FPGA[7]

- Operational **cloud microphysics** parameterisation
- FPGA (Maxeler MAX 5 DFE (Xilinx VU9P)): Automated **Fortran to C transpilation**, then hand ported to MaxJ
- GPU (NVidia Quadro GV100 Volta): **OpenACC offload directives**, optimized for GPU
- FPGA throughput **2.5x (3x) higher than CPU** and **2x (2.8x) higher than GPU** for DP (SP)



Outlook

- **Batch translation** of physical parameterisations
- **Fully automated** source-to-source translation of CLOUDSC to an FPGA version
- Extend to **other column based algorithms** in the IFS

1. Müller, A. *et al.* The ESCAPE project: Energy-efficient Scalable Algorithms for Weather Prediction at Exascale. *Geoscientific Model Development* **12,** 4425–4441 (2019).
2. Clement, V. *et al.* *The CLAW DSL.* in *Proceedings of the Platform for Advanced Scientific Computing Conference* (ACM, July 2018).
3. Klöckner, A. & Contributors. *Pymbolic.* https://github.com/inducer/pymbolic.
4. Open Fortran Project. *Open Fortran Parser.* http://fortran-parser.sourceforge.net.
5. Science and Technology Facilities Council. *fparser.* https://github.com/stfc/fparser.
6. Omni Compiler Project. *Omni Compiler.* https://omni-compiler.org.
7. Targett, J., Lange, M. & Marsden, O. *Investigation of FPGA implementation of operational cloud microphysics parameterisation.* in preparation.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 754337.