Imperial College London

Automated Code Generation in Fluidity

Christian Jacobs

Background

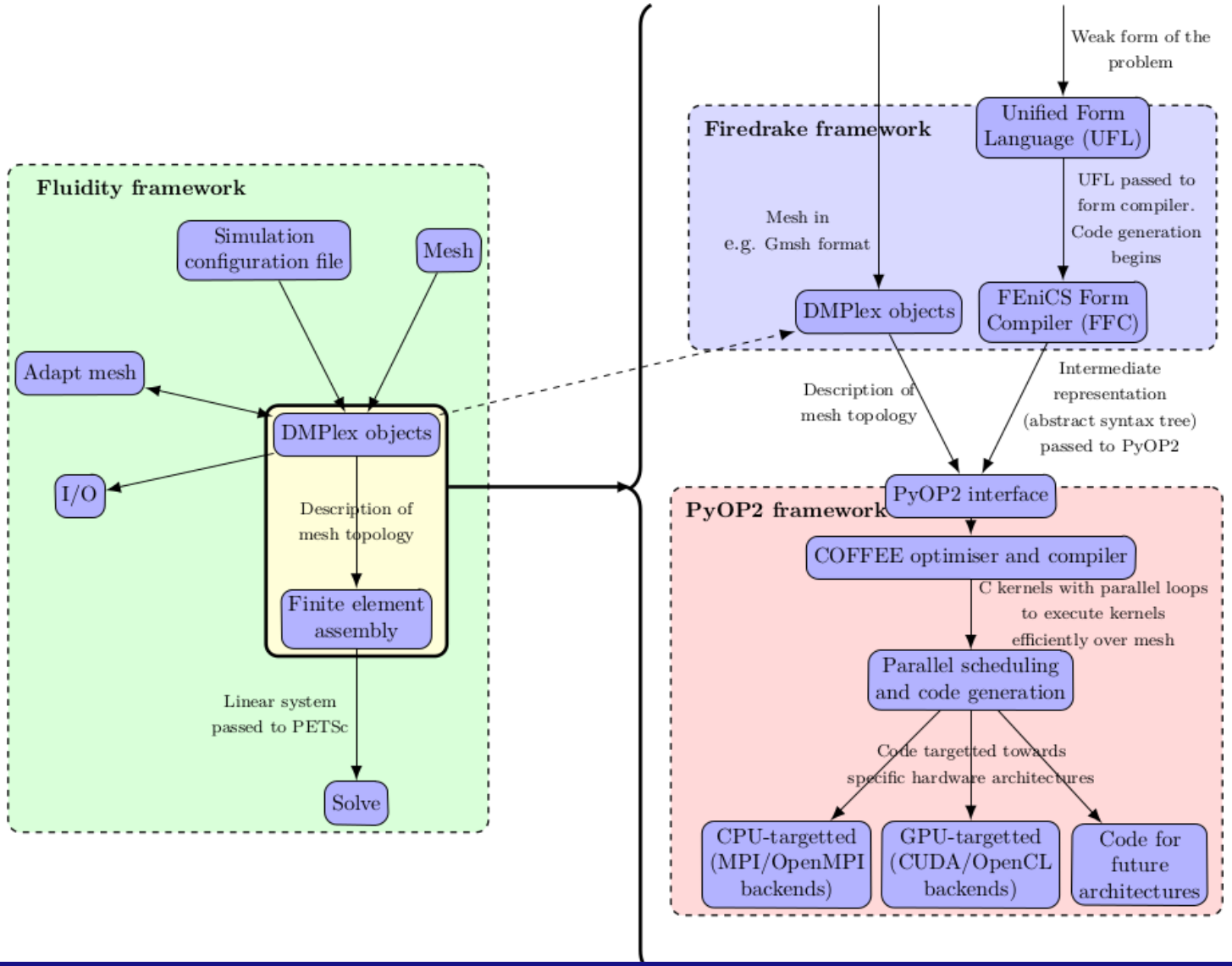
- The core of Fluidity comprises low-level, hand-written Fortran code to assemble the system of equations.
- This is typically sub-optimal and does not cater for different hardware, e.g. GPUs, AVX, ...
- We would need to re-engineer the hand-written code and throw in a few calls to CUDA, OpenCL or some other backend.
- This places extra burden on the developer to not only be an expert in numerical methods and their application area, but also an expert in software engineering and parallelisation.

Firedrake

- Firedrake (www.firedrakeproject.org) is a library which generates the assembly code automatically.
- Problems are specified in a high-level, near-mathematical, Python-based language called UFL...
- ...and then compiled down into optimised, low-level C code that is targetted towards a desired backend (MPI, MPI+OpenMP, CUDA, OpenCL, ...).

Re-engineering

- We are in the process of re-engineering Fluidity to use Firedrake's automated code generation techniques.
- Models and numerical schemes are being ported over from Fortran to UFL.


Fluidity framework
Simulation configuration file
Mesh
Adapt mesh
DMPlex objects
I/O
Description of mesh topology
Finite element assembly
Linear system passed to PETSc
Solve
Firedrake framework
Weak form of the problem
Unified Form Language (UFL)
UFL passed to form compiler. Code generation begins
Mesh in e.g. Gmsh format
DMPlex objects
FEniCS Form Compiler (FFC)
Description of mesh topology
Intermediate representation (abstract syntax tree) passed to PyOP2
PyOP2 framework
PyOP2 interface
COFFEE optimiser and compiler
C kernels with parallel loops to execute kernels efficiently over mesh
Parallel scheduling and code generation
Code targetted towards specific hardware architectures
CPU-targetted (MPI/OpenMPI backends)
GPU-targetted (CUDA/OpenCL backends)
Code for future architectures

Firedrake-Fluids

- A prototype for the 'new Fluidity' code which uses code generation, called Firedrake-Fluids, is available at:
  `github.com/firedrakeproject/firedrake-fluids`
- Currently only the shallow water model has been implemented in UFL, along with SU stabilisation and an LES turbulence model.

Using Code Generation

- Convert the old-style options file used with the ‘hand-written’ model to a ‘new-style’ one is compatible with the Firedrake-based model:
- `python tools/fl2ff.py old.flml new.swml`
- Execute the Firedrake-based model with the Python interpreter:
- `python firedrake_fluids/shallow_water.py new.swml`