COFFEE: an Optimizing Compiler for Finite Element Local Assembly

Fabio Luporini[2], Paul H. J. Kelly[2], and David A. Ham[1,2]

[1] *Department of Mathematics, Imperial College London*
[2] *Department of Computing, Imperial College London*

Local assembly is the characteristic operation of the finite element method, which entails the numerical evaluation of a problem-specific integral for each element in the discretized equation domain. Since the domain size can be huge, efficient computation of such integrals is fundamental. Many approaches are possible: in addition to the conventional quadrature representation, the introduction of DSLs and the adoption of run-time code generation has led to novel methods based on tensor contraction [1] and symbolic manipulation [2]. It has been demonstrated, however, that quadrature representation remains the optimal choice for a wide class of problems [3].

Low-level optimization of routines based on quadrature representation is, in general, a challenging issue. Even though an affine, typically non-perfect loop nest is generally present, the short trip counts and the complexity of mathematical expressions make it hard to determine a single or unique sequence of successful transformations. In this context, we present the design and systematic evaluation of COFFEE[1], a domain-specific compiler for local assembly integrated with the FEniCS-compatible system Firedrake. COFFEE manipulates abstract syntax trees produced by a modified version of the FEniCS Form Compiler (FFC), introducing composable optimizations aimed at improving instruction-level parallelism, especially SIMD vectorization, and register locality. It then generates C code including vector intrinsics. Experiments using a range of finite-element forms of increasing complexity show that significant performance improvement over FFC-optimized problems is achieved.

In this talk, we also mention two optimizations, which are currently under development in COFFEE, that are of general interest for the FEniCS community: inter-kernel vectorization and generation of specialised code for small linear algebra operations. The former enables SIMD vectorization in problems in which the amount of loop-invariant code, inherently unstructured, represents the bulk of the computation, for example those based on hyperelasticity; the latter centers on transforming the quadrature code into a sequence of calls to optimized BLAS routines.

[1] COFFEE stands for COmpiler For FinitE Element local assembly

References

[1] Robert C. Kirby and Anders Logg. A compiler for variational forms. *ACM Trans. Math. Softw.*, 32(3):417–444, September 2006.

[2] Francis P. Russell and Paul H. J. Kelly. Optimized code generation for finite element local assembly using symbolic manipulation. *ACM Transactions on Mathematical Software*, 39(4).

[3] Kristian B. Olgaard and Garth N. Wells. Optimizations for quadrature representations of finite element tensors through automated code generation. *ACM Trans. Math. Softw.*, 37(1):8:1–8:23, January 2010.