Enabling Manual Vectorization of Complex Code Patterns in Fortran

Florian Wende

Zuse Institute Berlin (ZIB)
Germany

What's unique about my tuning work

- **Generic approach to enable manual vectorization in Fortran**
  - Standard approach: "Array notation" AND/OR "Loops + Directives"
    - Limited expressiveness + no SIMD intrinsics
    - Compiler may fail to vectorize → Poor program performance?!
  - **Our approach:** SIMD-intrinsic-like coding in Fortran
    - Complex loops → Sequence of MACROS containing simple SIMD loops

```
Y=_mm512_add_epi32(X1,X2) → #define SIMD_ADD_INT32(Y,X1,X2)\
                                 do i=1,16;\
                                    Y%v(i)=X1%v(i)+X2%v(i);\
                                 enddo
```

    - Auto-vectorizer can fuse loops if meaningful
- **Application domain:** Fortran codes with complex loop structure
  - Test application: Connected component labeling (CCL) kernel
- **Tools used:** Intel Fortran compiler 15.0.0

Performance

- **Compelling performance:** We are almost as good as the compiler in cases where the standard approaches work
  - Our scheme should also work for complex loops which the auto-vectorizer cannot handle
- **Competitive performance:** C intrinsics version of CCL kernel about a factor 2 faster (Xeon Phi)
- **Speedups:** ~5x compared to non-vectorized code (Xeon Phi)
- **List of Optimizations:**
  - SIMD data types: SIMD width + Alignment
  - Fortran macros containing single-instruction loops
  - best performance for our CCL kernel with

```
ifort -O3 -mmic -align array64byte -opt-assume-safe-padding
```

Insights

- **What I learned:** Auto-vectorization (report) is a story by itself
- **What I recommend and how I would have done it differently:**
  - Design data layout for SIMD directly from the first
  - Test code sections for SIMD performance by porting them to "C + Intrinsics" (maybe not nice but effective)
- **Which tools/optimizations were most useful and why?**
  - Vec-report of the Intel 15 compiler: Infos about compiler assumptions regarding alignment, dependencies, etc.
  - Assembly code inspection
- **Biggest surprise:** C SIMD intrinsics version much faster
- **Key remaining challenges:** Apply scheme to large codes
- **Questions I'd like to raise:**
  - Why are there no SIMD intrinsics in Fortran?