ISC High Performance
The HPC Event.

BoF: Exploiting SIMD-Parallelism – Today & Tomorrow

How Effective is SIMD in Case of Divergent Code Execution?

Florian Wende

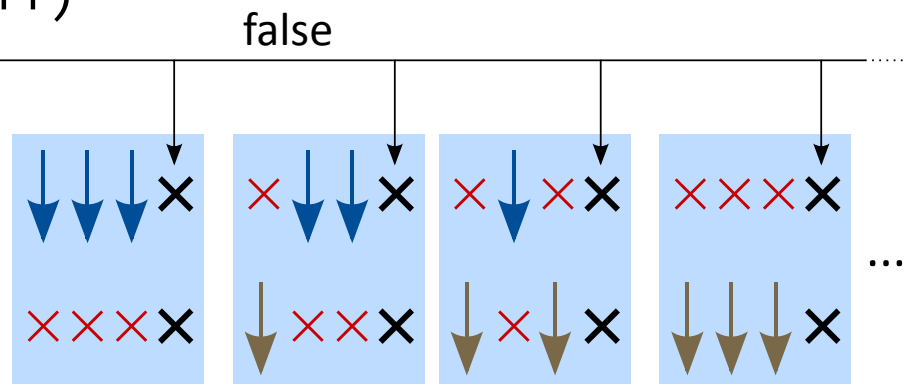
Zuse Institute Berlin

Divergences within SIMD loops

Caused by, e.g.,

- branching: `if-elseif-…-else`

```
for(int32_t i=0;i<LENGTH;i++)
  if(x[i]>0.0)
    if(x[i]<0.75)
      y[i]=exp(x[i]);
    else
      y[i]=log(x[i]);
```



How to address?

- vector masking [+ control logic]
- code restructuring, e.g., loop permutation for nested loops

Divergences within SIMD loops

Explicit vectorization with OpenMP 4.x SIMD directive

```
#pragma omp simd
for(int32_t i=0;i<LENGTH;i++){
  if(x[i]>0.0)
    if(x[i]<0.75)
      y[i]=exp(x[i]);
    else
      y[i]=log(x[i]);
}
```

SIMD intrinsics (Xeon Phi, IMCI)

```
for(int32_t i=0; i<LENGTH; i+=8){
  temp_1=_mm512_load_pd(&x[i]);
  mask_1=_mm512_cmp_pd_mask(temp_1,_mm512_set1_pd(0.0),_MM_CMPINT_GT);
  mask_2=_mm512_cmp_pd_mask(temp_1,_mm512_set1_pd(0.75),_MM_CMPINT_LT);
  temp_2=_mm512_load_pd(&y[i]);
  mask_3=_mm512_kand(mask_1,mask_2);
  mask_4=_mm512_kand(mask_1,_mm512_knot(mask_2));
  if(_mm512_mask2int(mask_3))
    temp_2=_mm512_mask_exp_pd(temp_2,mask_3,temp_1);
  if(_mm512_mask2int(mask_4))
    temp_2=_mm512_mask_log_pd(temp_2,mask_4,temp_1);
  _mm512_storenr_pd(&y[i],temp_2);
}
```

Explicit masking on Xeon Phi!

Will be available on CPUs with AVX-512 support, too

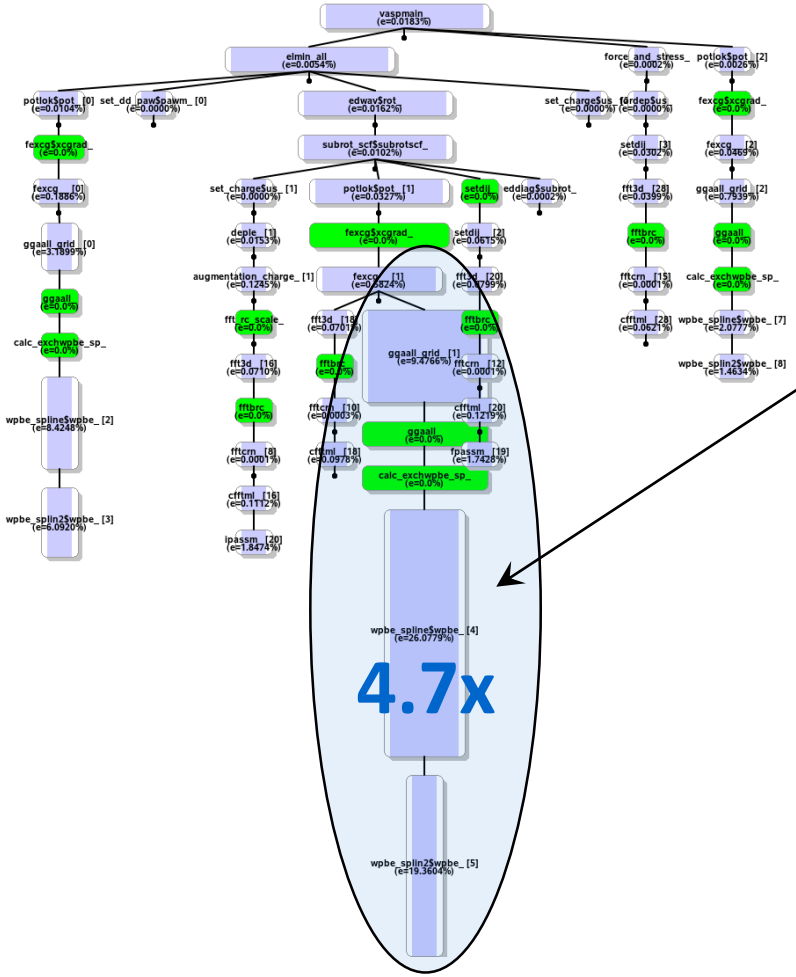
Execution on Xeon Phi 7120P (LENGTH=16x1024x1024, x[i]є[-1,+1], Intel 15.0.1 compiler: icpc -O3 -mmic -opt-assume-safe-padding -fp-model=precise)

Novector:	1004ms	
Pragma OMP SIMD:	**495ms**	
SIMD Intrinsics:	469ms	(-5.5%)

Performance with explicit vectorization close to intrinsics!

Divergences within SIMD loops – More Complex

Real-world code: VASP [Input: Methanol molecule in large volume]



Use of **SIMD-enabled functions**

- conditional (SIMD) function calls
- loops on SIMD lanes
- hundreds of lines of code

Explicit vectorization with OpenMP 4.x

- 1.5x on Xeon Phi with Fortran

Explicit vectorization + vector data types

- **4.7x** on Xeon Phi with Fortran

Manual vectorization

- 5.8x on Xeon Phi with C SIMD Intrinsics

Divergences within SIMD loops – Insights

SIMD can be effective in case of divergent code execution

Try auto-vectorizer and compiler directives: it closes up with SIMD intrinsics in many cases

Best performance with SIMD intrinsics in complex cases

Use OpenMP 4.x SIMD directive!

for portability reasons

Thank You