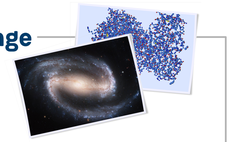
Watch your language! Modernizing HPC Software with C++11

I. Kabadshow, A. Beckmann, D. Haensel
Jülich Supercomputing Centre, Research Centre Jülich, Germany

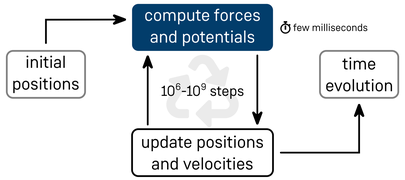
JÜLICH Forschungszentrum

⓪ Scientific Challenge



- Molecular Dynamics
- Astrophysics

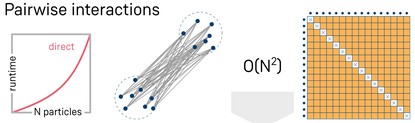
Method: Compute all pairwise interactions of N particles



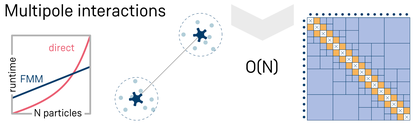
Problem: $O(N^2)$ complexity limits particle system size

① Algorithmic Solution

Pairwise interactions



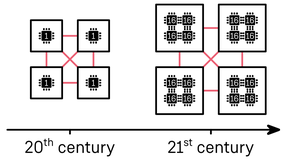
Multipole interactions



② Parallelization Potential

③ HPC Ecosystem - Then and Now

Hardware

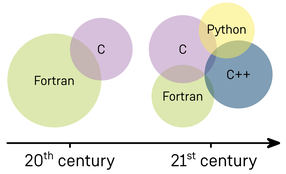


Vectorization
Threading
Message Passing

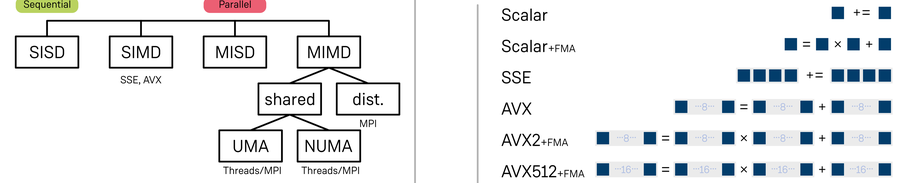
- Vectorization holds large amount of parallelization potential
- ! Compiler doesn't vectorize non-trivial code automatically

- Threading holds large amount of parallelization potential
- ! Fork/join parallelism may not be sufficient

Programming Languages



④ SIMD Vectorization

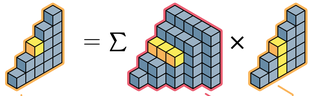


⑤ SIMD 101

- Consecutive, aligned memory for load/store needed
- Misaligned memory access slower or not allowed
- Not all operations are available (trig, exp, log)
- Some operations may be available as estimates
- ALU bandwidth/latency depends on operation

⑥ Portable Vectorization?

- ◎ Write the compute kernel once (high-level C++11)
- Reuse it with different precisions (float, double)
- Reuse it on different SIMD widths
- ! What about non-trivial datastructures?



$$\mu'_{lm} = \sum_{k=-l}^{l} \sqrt{\frac{(l-m)!(l+m)!}{(l-k)!(l+k)!}} \cdot d_l^{mk}(\theta) \cdot e^{im\phi} \cdot \mu_{lk}$$

- Naive approach: vectorize innermost loop
- ! Needs padding to SIMD width or special handling

⑦ Utilizing Resources Better

- Reuse rotation operator d
- Stack triangles element-wise: AoS to SoA
- Use temporary reordering/construction of in/output
  - Permanent storage in stacked layout not useful
  - Only single, fixed permutation storable
- Create triangular array of "complex" stacks
  - Arithmetic operations on stacks element-wise

⑧ Original Datalayout

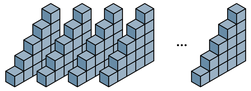
Complex

R I

Struct of Complex (SoC)

R I R I R I

Array of Struct of Complex (AoSoC)



⑨ Modified Datalayout

2x Array of Real

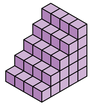
R R R R I I I I

Complex of Array of Real

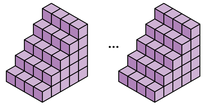
R R R R I I I I

Struct of Complex of Array (SoCoA)

R R R R I I I I R R R R I I I I



Array of Struct of Complex of Array (AoSoCoA)



Ⓐ SIMD-Portable Code

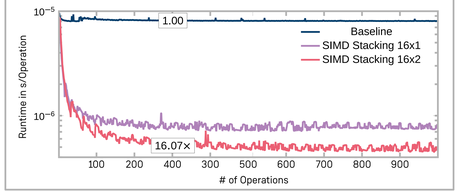
- Portable source code for float/double
- Portable source code for scalar/SSE/AVX/AVX2/AVX512

```
template <typename TA_in, typename Rot, typename TA_out>
void RotationLocalBackward(
    const TA_in &mu_in_rot, const Rot &R, TA_out &mu_out, const size_t p)
{
    for (size_t l = 0; l <= p; ++l) {
        for (size_t m = 0; m <= l; ++m) {
            typename TA_out::value_type mu = 0.0;
            for (size_t k = 1; k <= l; ++k)
                mu += scale_complex(R.d_g(l, m, k), mu_in_rot(l, k));
            mu *= R.e_imphi[m];
            mu_out(l, m) += mu;
        }
    }
}
```

Ⓑ Compiler-Generated Assembly

- AVX512@KNL: Intel Xeon-Phi-7250F, float, GCC 7.2.0
- Abstraction can handle arbitrary SIMD widths

```
loop:
vbroadcastss 0x4(%rcx,%rdx,8),%zmm0
lea    0x100(%rax),%rax
vbroadcastss (%rcx,%rdx,8),%zmm1
lea    0x1(%rdx),%rdx
cmp    %r10,%rdx
vfmadd231ps -0x100(%rax),%zmm1,%zmm5
vfmadd231ps -0xc0(%rax),%zmm1,%zmm2
vfmadd231ps -0x80(%rax),%zmm0,%zmm4
vfmadd231ps -0x40(%rax),%zmm0,%zmm3
jle    loop
```

Ⓒ Performance

- ! Without reordering costs, float, p=10