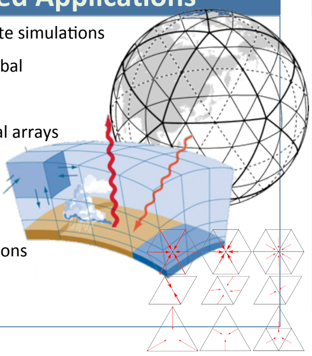
GridTools: A C++ Library for Computations on Grids

Mauro Bianco, Paolo Crosetto, Oliver Fuhrer,
Stefan Moosbrugger, Carlos Osuna, Hannes Vogt, Thomas C. Schulthess

Targeted Applications

- ✓ Weather and climate simulations
  - ✓ Regional and global
- ✓ Regular Grids
  - ✓ Multidimensianal arrays
- ✓ Structured grids
  - ✓ Icosahedral
  - ✓ A- and C-grids
  - ✓ Regular tasselations
  - ✓ Possibly others

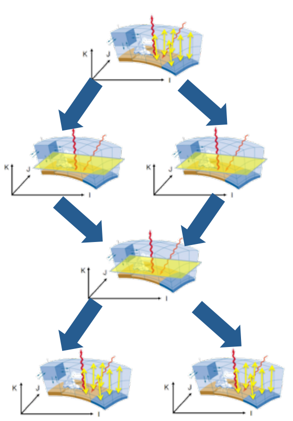


Properties & Requirements

- ✓ **Regular grids** and **stencils** are main motifs
- ✓ Stencils are typically memory **bandwidth bound**
- ✓ Applications run **many stencils** in sequence
- ✓ Compilers find these problems **hard to optimize**
- ✓ Typical time-loop optimizations are unpractical
- ✓ **Portability of performance**
- ✓ **Maintainability** and **separation of concerns**
  - ✓ Between application developers and performance specialists

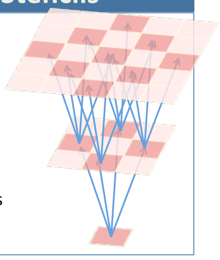
GridTools Ecosystem

Stencil Dependencies



Multi-Stage Stencils

- ✓ GridTools is for complex problems
- ✓ **Operator Splitting** for
  - ✓ Tractability
  - ✓ Maintainability
- ✓ **Dependencies** b/w operators
- ✓ **Time and space** dependencies
- ✓ GridTools provides easy composition of the stages



Stencil Operator

```
struct lap_function {
  typedef inout_accessor<0> out;
  typedef in_accessor<1, extent<-1,1,-1,1> > in;
  typedef arg_list = make_arg_list<out, in>;

  template <typename Evaluator>
  static void Do(Evaluator& eval) {
    eval(out()) = eval(4*in() -
                  in( 1, 0, 0) + in( 0, 1, 0) +
                  in(-1, 0, 0) + in( 0,-1, 0));
  }
};
```

Traditional Approach

```
for i=bi-1, ei+1
 for j=bj-1, ej+1
  for k=bk, ek
    lap(i,j,k) = 4*in(i,j,k) -
          (in(i+1,j,k) + in(i,j+1,k) +
           in(i-1,j,k) + in(i,j-1,k))

for i=bi-1, ei
 for j=bj, ej
  for k=bk, ek
    flx(i,j,k) = lap(i+1,j,k)-
lap(i,j,k);
     if (flx(i,j,k)*(in(i+1,j,k)-
in(i,j,k)))
       then flx(i,j,k) = 0.

for i=bi, ei
 for j=bj-1, ej
  for k=bk, ek
    fly(i,j,k) = lap(i,j+1,k)-
lap(i,j,k);
     if (fly(i,j,k)*(in(i,j+1,k)-
in(i,j,k)))
       then fly(i,j,k) = 0.

for i=bi, ei
 for j=bj, ej
  for k=bk, ek
    out(i,j,k) = in(i,j,k) -
      (flx(i,j,k) - flx(i-1,j,k) +
       fly(i,j,k) - fly(i,j-1,k))
```

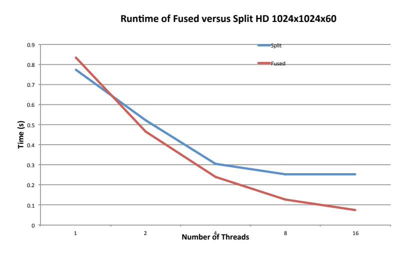
Multi-Stage Stencil Composition

```
auto horizontal_diffusion =
  make_computation<BACKEND> (
    make_multistage(
      execute<forward>,
      make_stage<lap_f>(lap(), in()),
      make_independent(
        make_stage<flx_f>(flx(),in(),lap()),
        make_stage<fly_f>(fly(),in(),lap())),
      make_stage<comb_f>(out(),in(),flx(),fly())),
    data_fields, coords);

horizontal_diffusion->run();
```

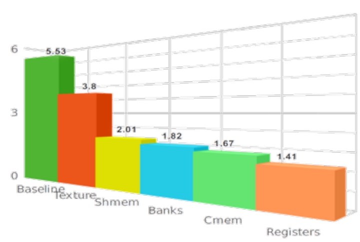
Grid-Independent Operator

```
template < uint_t Color > struct sum_on_cells {
  using in = in_accessor<0, cells, extent<-1,1,-1,1>>;
  using out = inout_accessor<1, cells>;
  using arg_list = make_arg_list<in, out>;

  template < typename Evaluator >
  static void Do(Evaluator& eval) {
    eval(out()) = eval(on_cells(
            [](double _i, double _r) {return _i+_r;},
            0.0, in()));
  }
};
```

Impact of Fusion



Incremental Optimization



Contact

Mauro Bianco
Swiss National Supercomputing Centre
Email: mbianco@cscs.ch
Phone: +41 (0) 91 610 8279

References

1. T. Gysi, O. Fuhrer, C. Osuna, M. Bianco, and T. Schulthess. Stella: A domain-specific language and tool for structured grid methods. Submitted.
2. M. Bianco, B. Cumming. A generic strategy for multi-stage stencils. EuroPar'14, Aug. 25-29 2014, Porto, Portugal.
3. M. Bianco, U. Varetto. A generic library for stencil computations. CoRR, abs/1207.1746, 2012.
4. J. Järvi, M. Marcus, J. Smith. Library composition and adaptation using C++ concepts. Proceeding of 6th International Conference on Generative Programming and Component Engineering, GPCE, 2007, Salzburg, Austria.
5. G. Dos Reis, J. Järvi. What is generic programming? In Proceedings of the First International Workshop of Library-Centric Software Design (LCSD '05). An OOPSLA 2005 workshop. San Diego, California, USA, October 2005.
6. M. Mernik, J. Heering, A. Sloane. When and how to develop domain-specific languages, ACM Computing Surveys, December 2005, ACM

Platform for Advanced Scientific Computing