





#### **Applications : recent progress at CSCS**

Joost VandeVondele, CSCS February, 2018





#### **Gridtools on KNL**

#### **Gridtools on KNL**

Goal:

KNL backend for the Gridtools library

Done:

- Performance evaluation and optimization of stencils on KNL
  - Blocking decisions
  - Loop ordering decisions
  - Hard-/Software pre-fetching
  - 'auto'-vectorization
- Basic KNL backend in Gridtools (still work in progress)

Future:

Good learning experience, carries over to future processors

In collaboration with Intel.



How hard can it be (I) ?

i-j-k Layout



i-k-j Layout

Block size sensitivity, depending on loop layout



#### How hard can it be (II) ?



#### Vertical Advection

Optimal hyper-threading, depending on stencil type





#### **Performance Optimization**

#### Horizontal Diffusion

#### **Vertical Advection**







#### **Backend Performance**

Old CPU backend (128×128×80)

New KNL backend (128×128×80)



Up to 10x speedup on KNL. Significant gain on Skylake.









#### **Task based programming**

## HPX for HPC on a node

- library for task based programming, well integrated in C++
- Cholesky decomposition matches Parsec
- Increasingly adding features for HPC (resource partitioning, schedule refinements)



## HPX : towards composability

Work on allowing a suspend/resume feature, essential for libraries.







## Linear Algebra : communication optimality

# Communication Optimal Matrix Multiplication

Adapt and implement the CARMA algorithm.

CARMA is the first algorithm that is shown to be **communication optimal** for **any** matrix dimensions and **all** memory ranges





James Demmel et. al (2013), Communication-Optimal Parallel Recursive Rectangular Matrix Multiplication, IEEE 27<sup>th</sup> Symposium on Parallel and Distributed Processing

#### Towards CARMA in production..



In progress

In progress

Goals:

- Find the initial data layout for the existing implementation
- Test and fix the existing implementation
- Make it work for any matrices dimensions and number processors (not just powers of 2)
  - Add a compatibility layer to allow other data layouts as input
- Optimize:
  - Allocate buffers only when necessary
  - Overlap communication and computation
  - Use GPU for additional acceleration
- Experiment with different underlying CARMA data layout without loosing the performance

Verified literature results, increasing usability







#### Thank you for your attention.