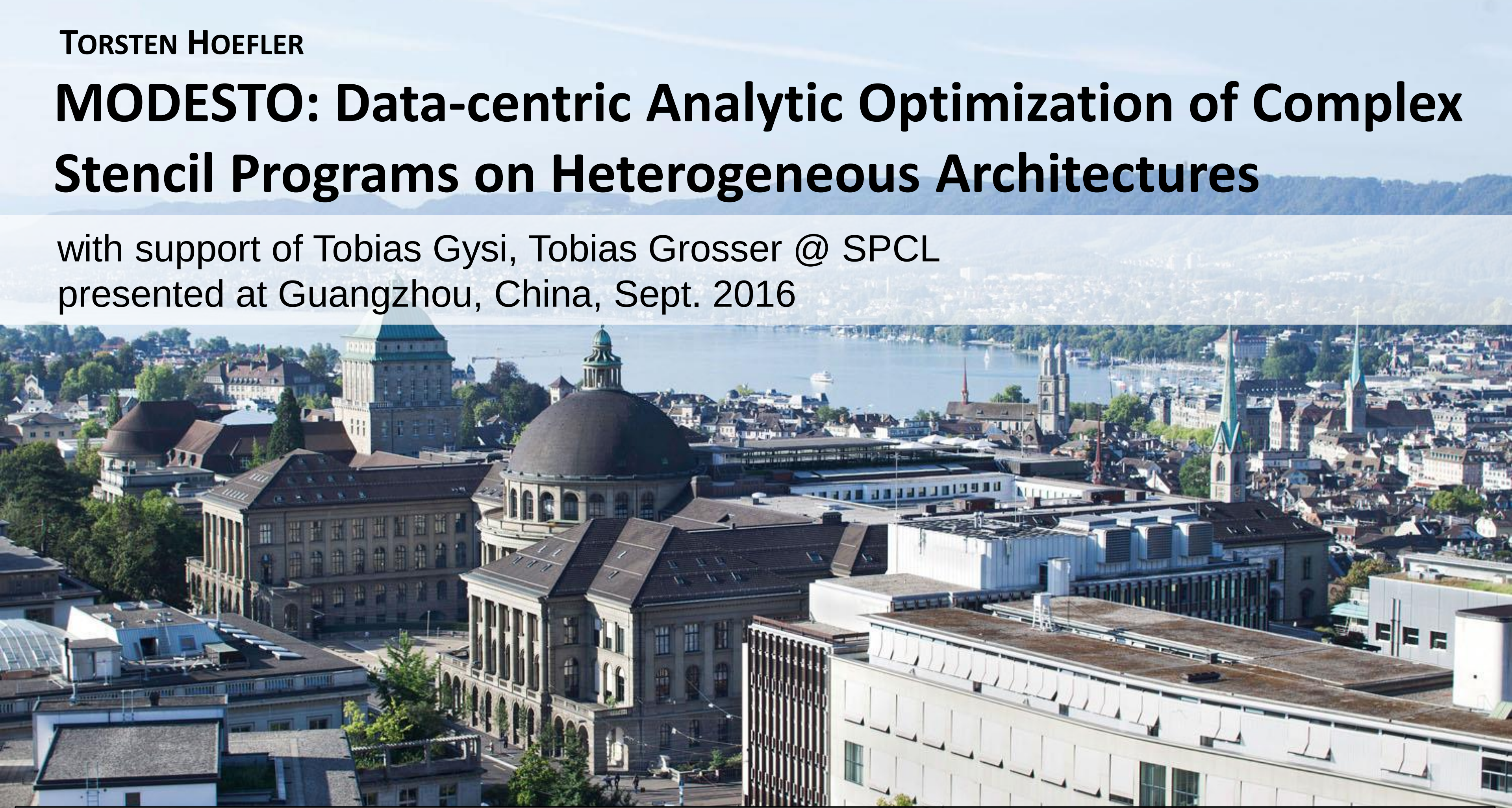
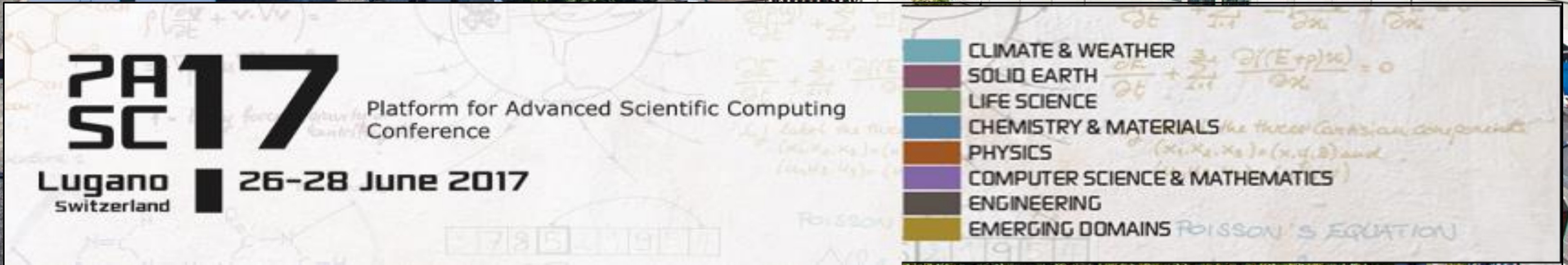
TORSTEN HOEFLER

MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures

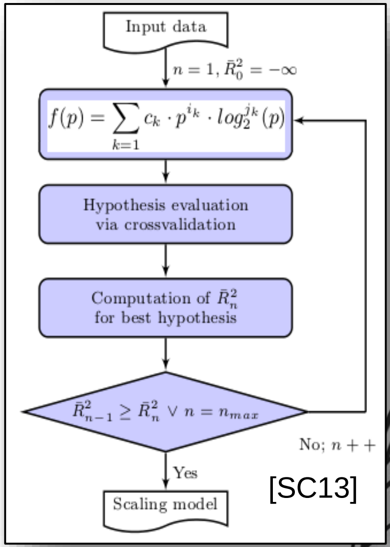
with support of Tobias Gysi, Tobias Grosser @ SPCL
presented at Guangzhou, China, Sept. 2016





ETH, CS, Systems Group, SPCL

- **ETH Zurich – top university in central Europe**
  - Shanghai ranking '15 (Computer Science): #17, best outside North America
  - 16 departments, 1.62 Bn $ federal budget
  - CSCS is part of ETH (houses Europe's fastest HP machine)
- **Computer Science department**
  - 28 tenure-track faculty, 1k students
- **Systems group (7 professors)**
  - O. Mutlu, T. Roscoe, G. Alonso, A. Singla, C. Zheng, D. Kossmann, TH
  - Focused on systems research of all kinds (data management, OS, …)
- **SPCL focusses on performance/data/HPC**
  - 1 faculty
  - 3 postdocs
  - 8 PhD students (+2 external)
  - 15+ BSc and MSc students
  - http://spcl.inf.ethz.ch
  - Twitter: @spcl_eth


Input data
$n = 1, \bar{R}_0^2 = -\infty$
$f(p) = \sum_{k=1} c_k \cdot p^{i_k} \cdot log_2^{j_k}(p)$
Hypothesis evaluation via crossvalidation
Computation of $\bar{R}_n^2$ for best hypothesis
$\bar{R}_{n-1}^2 \geq \bar{R}_n^2 \vee n = n_{max}$
No; $n++$
Yes
Scaling model

[SC13]

OpenSM

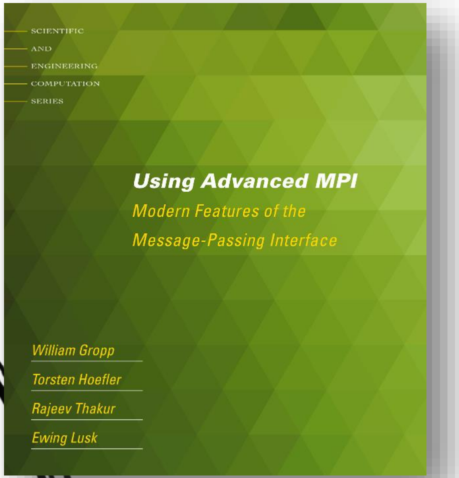
DFSSSP

Performance Modeling

DAPP

Parallel Programming

Large-Scale Networking



Scientific and Engineering Computation Series

Using Advanced MPI

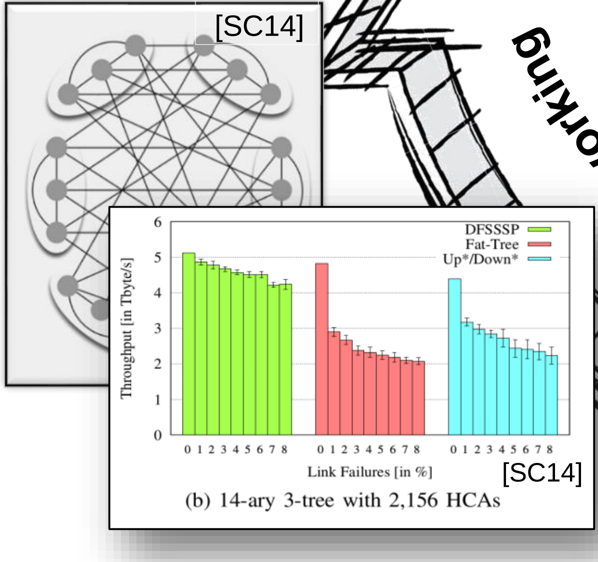
Modern Features of the Message-Passing Interface

William Gropp

Torsten Hoefler

Rajeev Thakur

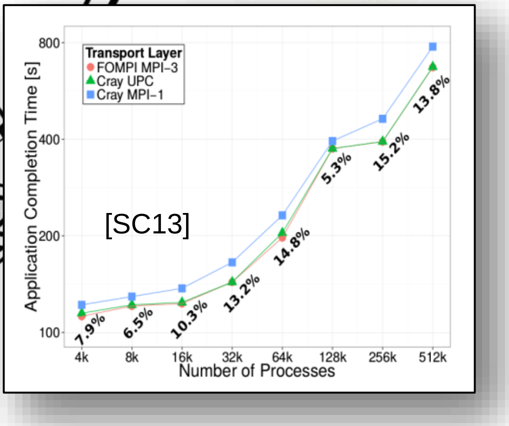
Ewing Lusk



[SC14]

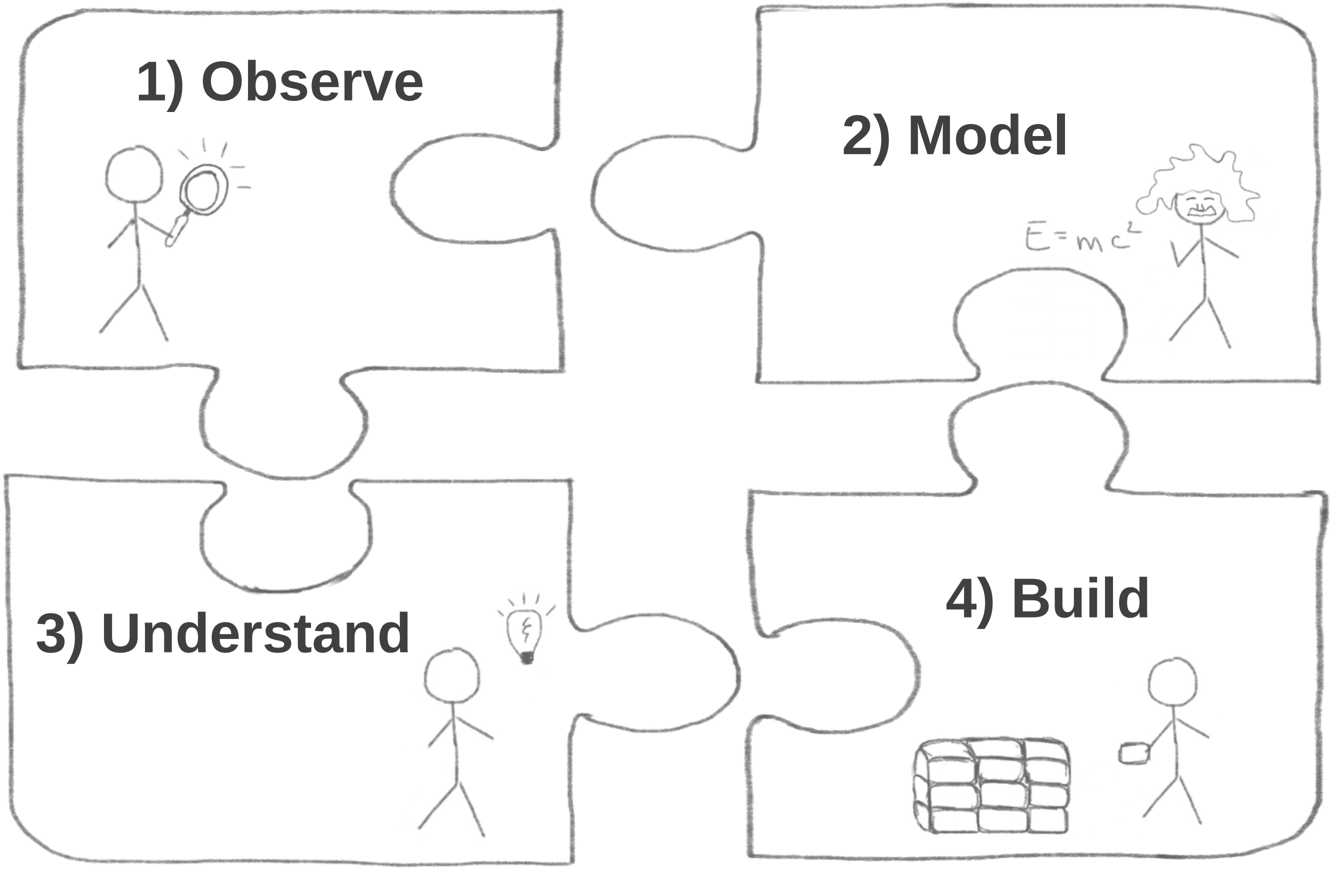
(b) 14-ary 3-tree with 2,156 HCAs

[SC14]



[SC13]

Scientific Performance Engineering



1) Observe

2) Model

3) Understand

4) Build

More at keynote at HPC China next month in Xi’an!

Stencil computations (oh

due to their low arithmetic intensity stencil computations are typically heavily memory bandwidth limited!

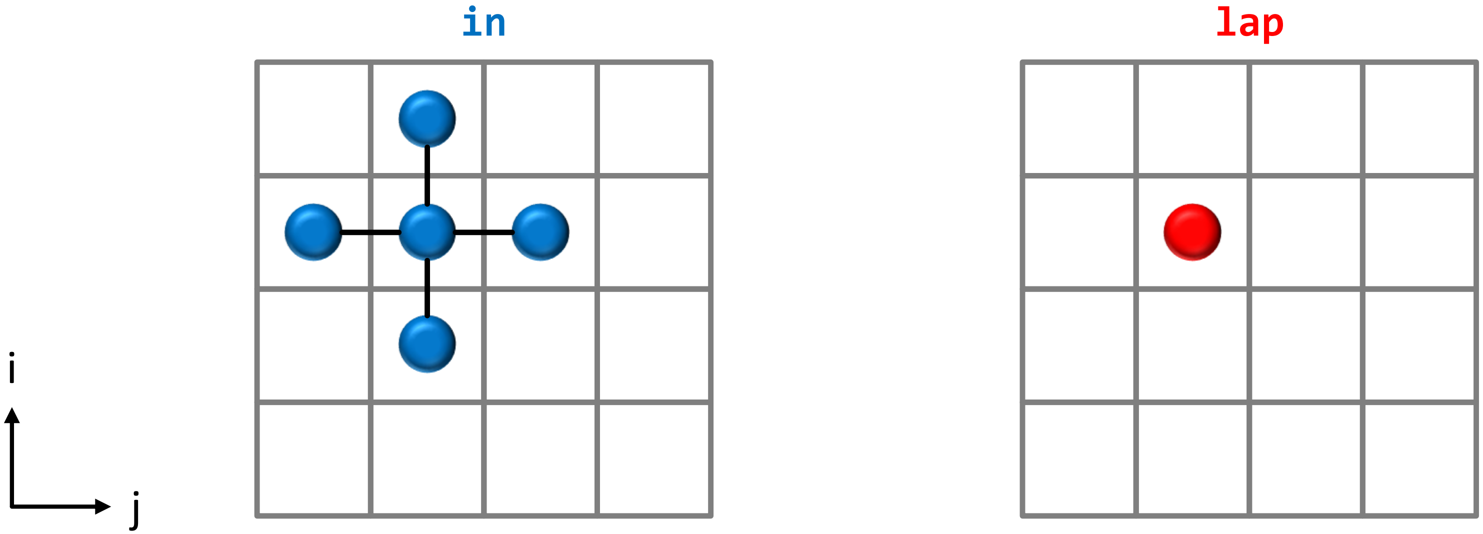
Motivation:

- **Important algorithmic motif (e.g., finite difference method)**

Definition:

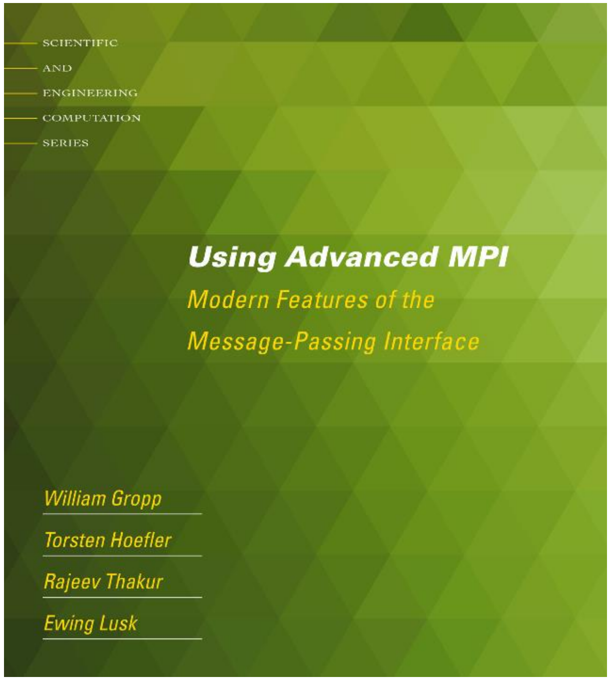
- **Element-wise computation on a regular grid using a fixed neighborhood**
- **Typically working on multiple input fields and writing a single output field**

```
lap(i,j) = -4.0 * in(i,j) + in(i-1,j) + in(i+1,j) + in(i,j-1) + in(i,j+1)
```



How to tune such stencils (most other stencil talks)

- **LOTS of related work!**
  - Compiler-based (e.g., Polyhedral such as PLUTO [1])
  - Auto-tuning (e.g., PATUS [2])
  - Manual model-based tuning (e.g., Datta et al. [3])
  - … essentially every micro-benchmark or tutorial, e.g.:



SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES

Using Advanced MPI

Modern Features of the Message-Passing Interface

William Gropp
Torsten Hoefler
Rajeev Thakur
Ewing Lusk

- **Common features**
  - Vectorization tricks (data layout)
  - Advanced communication (e.g., MPI neighbor colls)
  - Tiling in time, space (diamond etc.)
  - Pipelining

- **Much of that work DOES NOT compose well with practical complex <u>stencil programs</u>**

[1]: Uday Bondhugula, A. Hartono, J. Ramanujan, P. Sadayappan. *A Practical Automatic Polyhedral Parallelizer and Locality Optimizer* , PLDI’08
[2]: Matthias Christen, et al.: *PATUS: A Code Generation and Autotuning Framework for Parallel Iterative Stencil Computations …, IPDPS’11*
[3]: Kaushik Datta, et al., *Optimization and Performance Modeling of Stencil Computations on Modern Microprocessors,* SIAM review

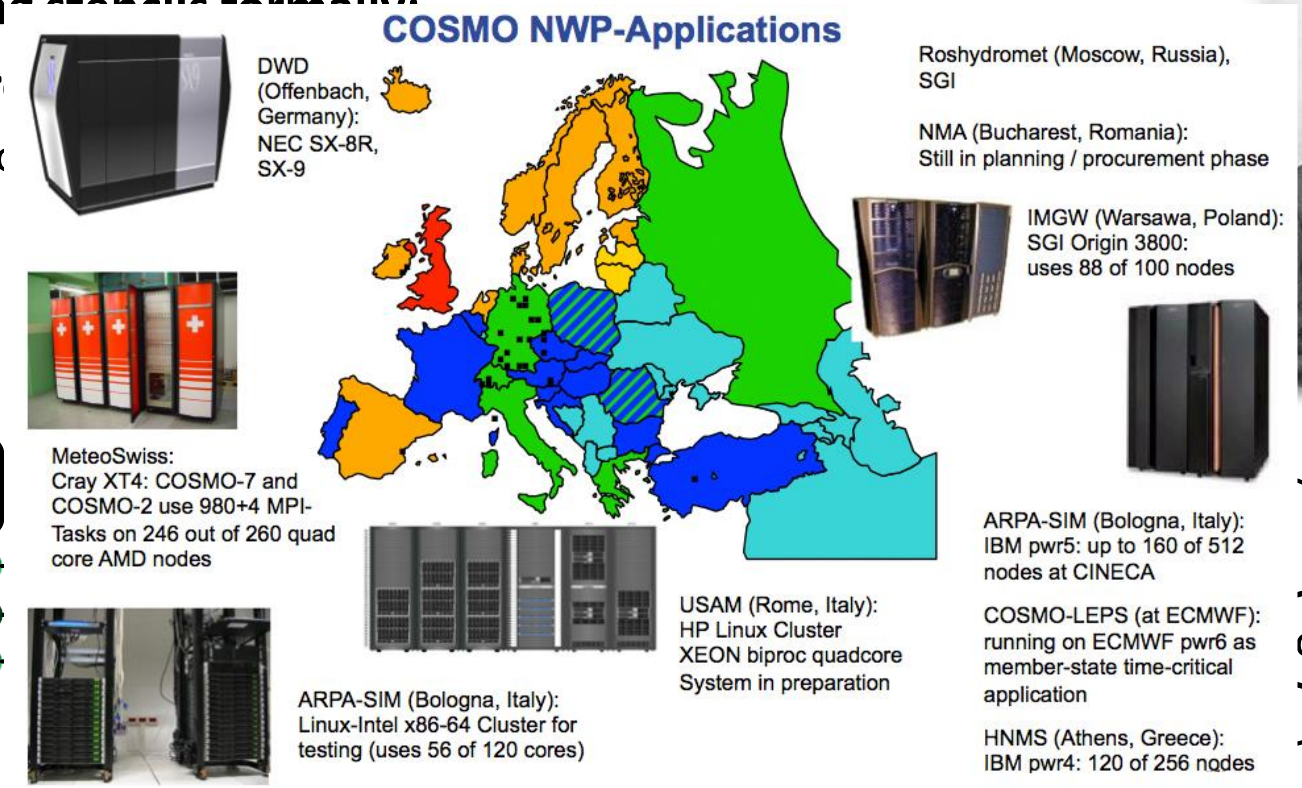
What is a “complex stencil program”? (this stencil talk)

E.g., the COSMO weather code

- **is a regional climate model used by 7 national weather services**
- **contains hundreds of different complex stencils**

Modelin

- **Repr**
  - Mo



COSMO NWP-Applications

DWD (Offenbach, Germany): NEC SX-8R, SX-9

Roshydromet (Moscow, Russia), SGI

NMA (Bucharest, Romania): Still in planning / procurement phase

IMGW (Warsawa, Poland): SGI Origin 3800: uses 88 of 100 nodes

MeteoSwiss: Cray XT4: COSMO-7 and COSMO-2 use 980+4 MPI-Tasks on 246 out of 260 quad core AMD nodes

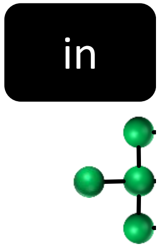
ARPA-SIM (Bologna, Italy): IBM pwr5: up to 160 of 512 nodes at CINECA

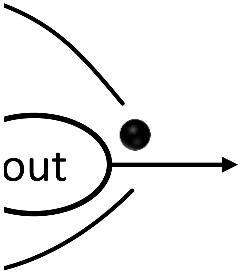
USAM (Rome, Italy): HP Linux Cluster XEON biproc quadcore System in preparation

COSMO-LEPS (at ECMWF): running on ECMWF pwr6 as member-state time-critical application

ARPA-SIM (Bologna, Italy): Linux-Intel x86-64 Cluster for testing (uses 56 of 120 cores)

HNMS (Athens, Greece): IBM pwr4: 120 of 256 nodes

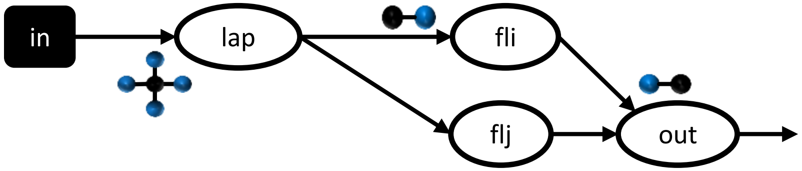




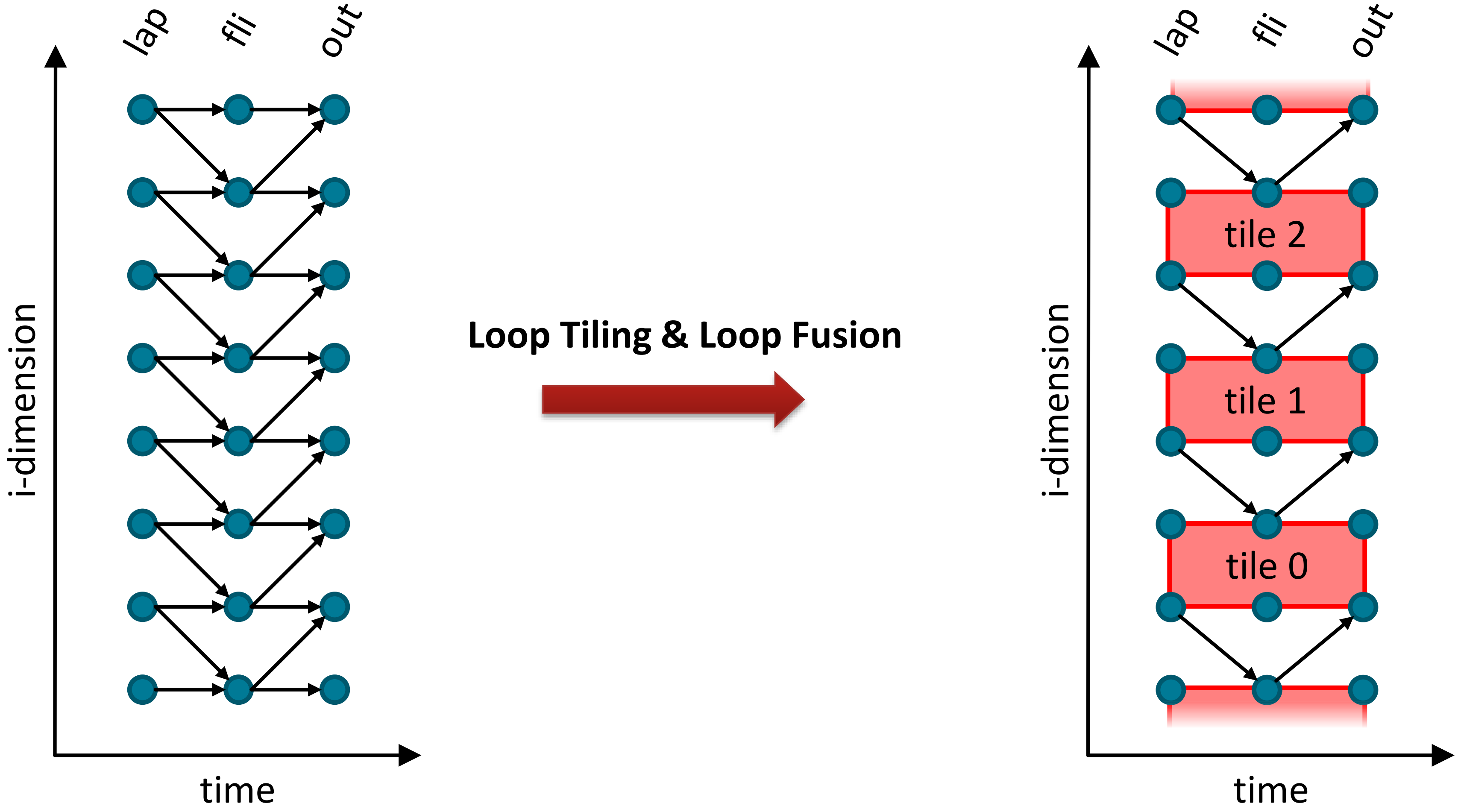
$$a \oplus b = \{a' + b' \mid a' \in a, b' \in b\}$$

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS’15

Data-locality Transformations



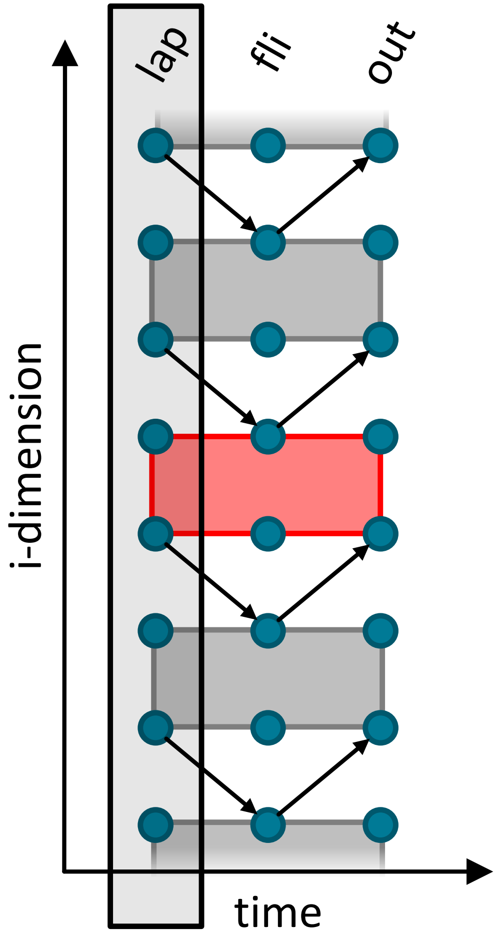
- Consider the horizontal diffusion lap-fli-out dependency chain (i-dimension)



T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

How to Deal with Data Dependencies?

- Consider the horizontal diffusion lap-fli-out dependency chain (i-dimension)



Halo Exchange Parallel (hp):

- Update tiles in parallel
- Perform halo exchange communication

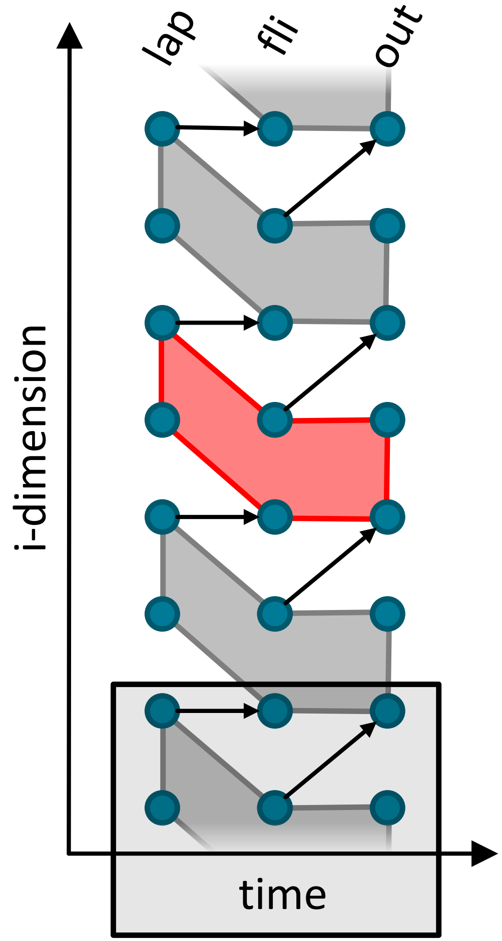
Pros and Cons:

- Avoid redundant computation
- At the cost of additional synchronization

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

How to Deal with Data Dependencies?

- Consider the horizontal diffusion lap-fli-out dependency chain (i-dimension)



Halo Exchange Sequential (hs):

- Update tiles sequentially
- Innermost loop updates tile-by-tile

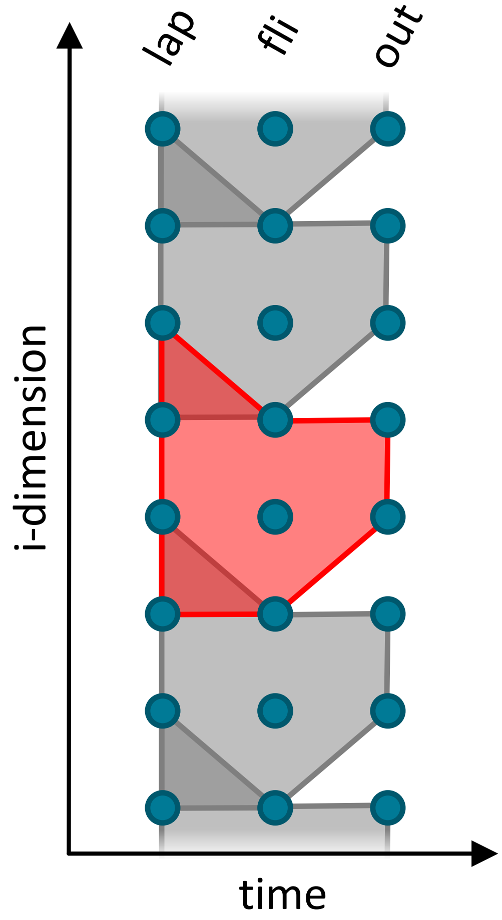
Pros and Cons:

- Avoid redundant computation
- At cost of being sequential

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

How to Deal with Data Dependencies?

- **Consider the horizontal diffusion lap-fli-out dependency chain (i-dimension)**



Computation on-the-fly (of):

- Compute all dependencies on-the-fly
- Overlapped tiling

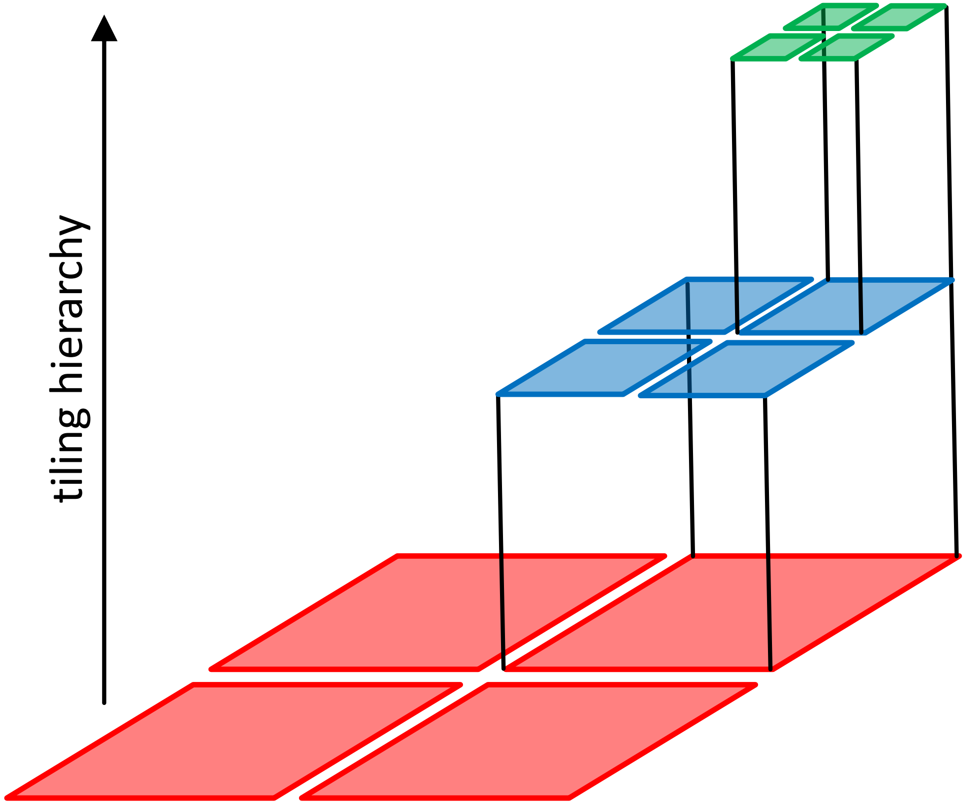
Pros and Cons:

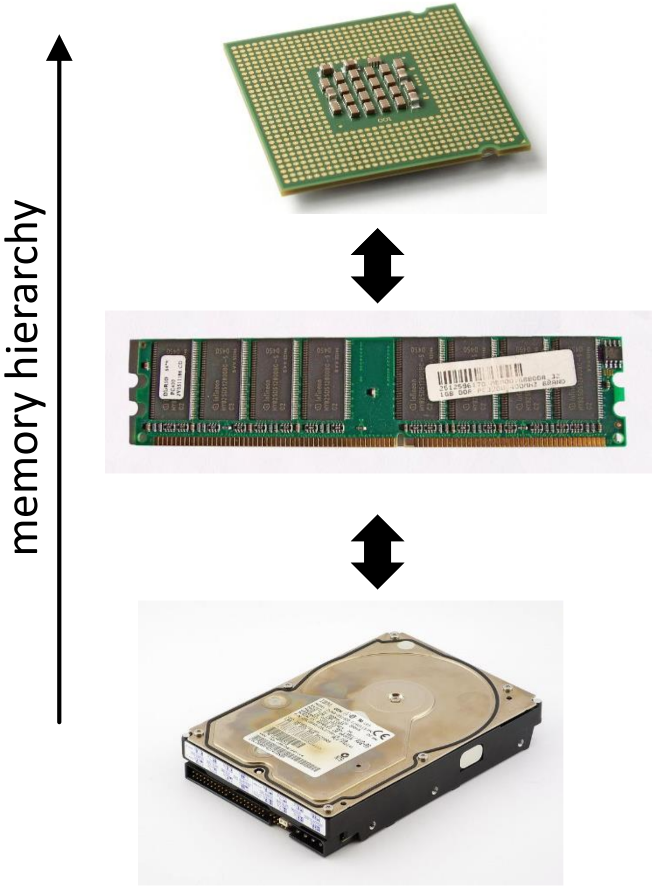
- Avoid synchronization
- At the cost of redundant computation

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Hierarchical Tiling

- **By tiling the domain repeatedly we target multiple memory hierarchy levels**





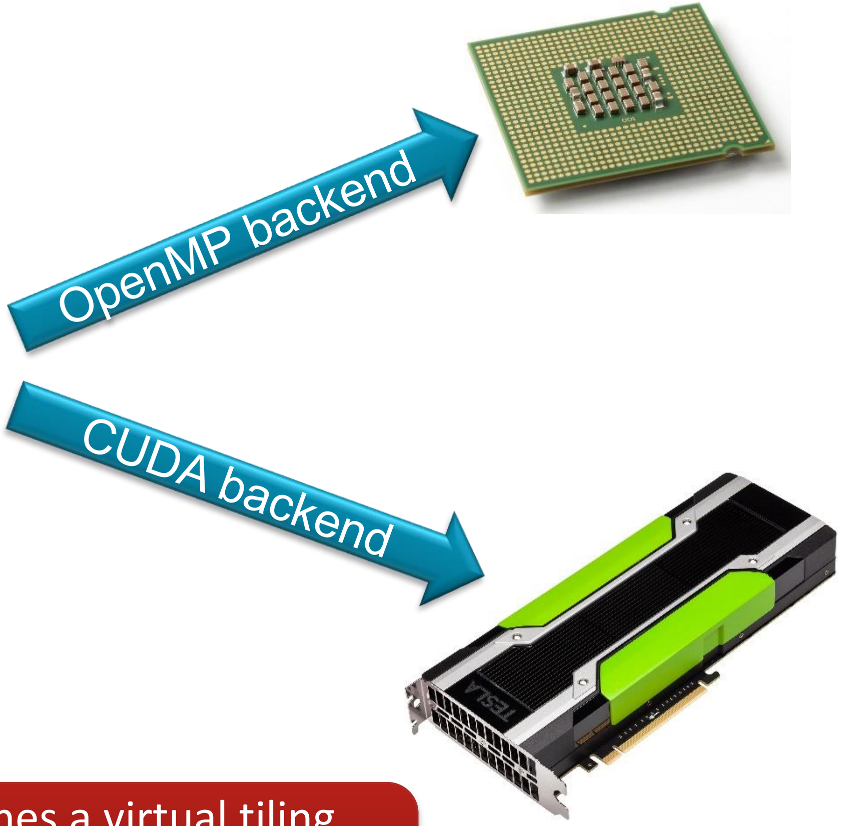
T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Case Study: STELLA (STEncil Loop LAnguage)

- STELLA is a C++ stencil DS(e)L of COSMO's dynamical core (50k LOC, 60% RT)

```
// define stencil functors
struct Lap { ... };
struct Fli { ... };
...
// stencil assembly
Stencil stencil;
StencilCompiler::Build(
 stencil,
 pack_parameters( ... ),
 define_temporaries(
  StencilBuffer<lap, double>(),
  StencilBuffer<fli, double>(),
  ...
 ),
 define_loops(
  define_sweep(
   StencilStage<Lap, IJRange<-1,1,-1,1> >(),
   StencilStage<Fli, IJRange<-1,0,0,0> >(),
   ...
)));
// stencil execution
stencil.Apply();
```

using C++ template metaprogramming:

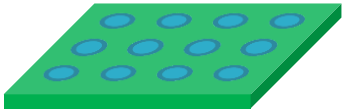
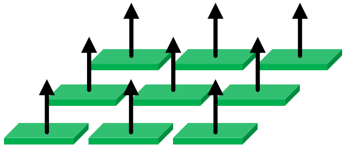
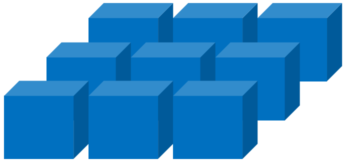


STELLA defines a virtual tiling hierarchy that facilitates platform independent code generation

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Tiling Hierarchy of STELLA’s GPU-Backend

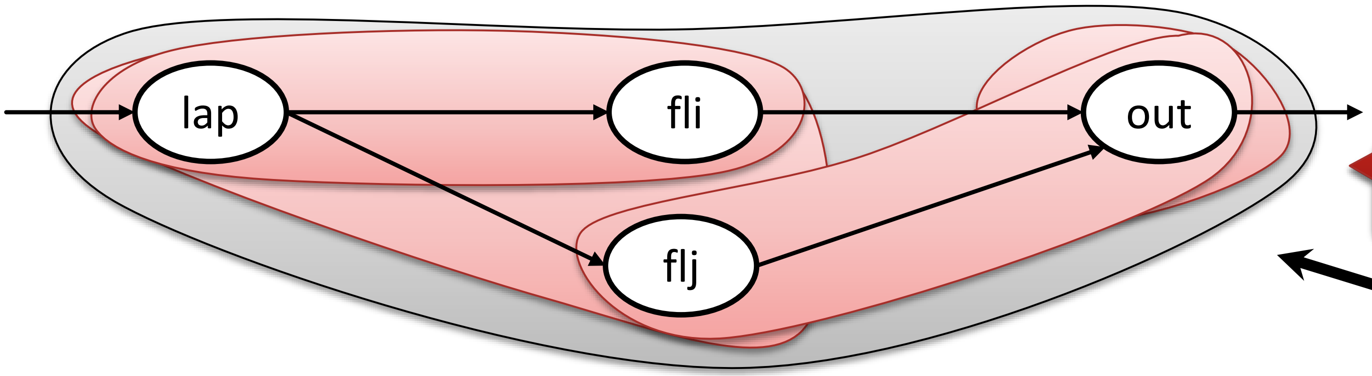
tiling hierarchy

DSL	Tile Size	Strategy	Memory	Communication	
sweep	1 x 1 x 1	halo exchange parallel	registers	scratchpad	
sweep	∞ x ∞ x 1	halo exchange sequential	registers	registers	
loop	64x4x64	computation on-the-fly	GDDR	-	
stencil	∞ x ∞ x ∞	computation on-the-fly	GDDR	-	

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS’15

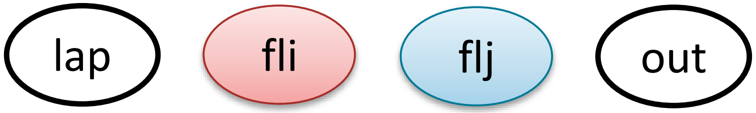
Stencil Program Algebra

- Map stencils to the tiling hierarchy using a bracket expression



[[lap,fli,flj],[out]]

- Enumerate the stencil execution orders that respect the dependencies



- Enumerate implementation variants by adding/removing brackets

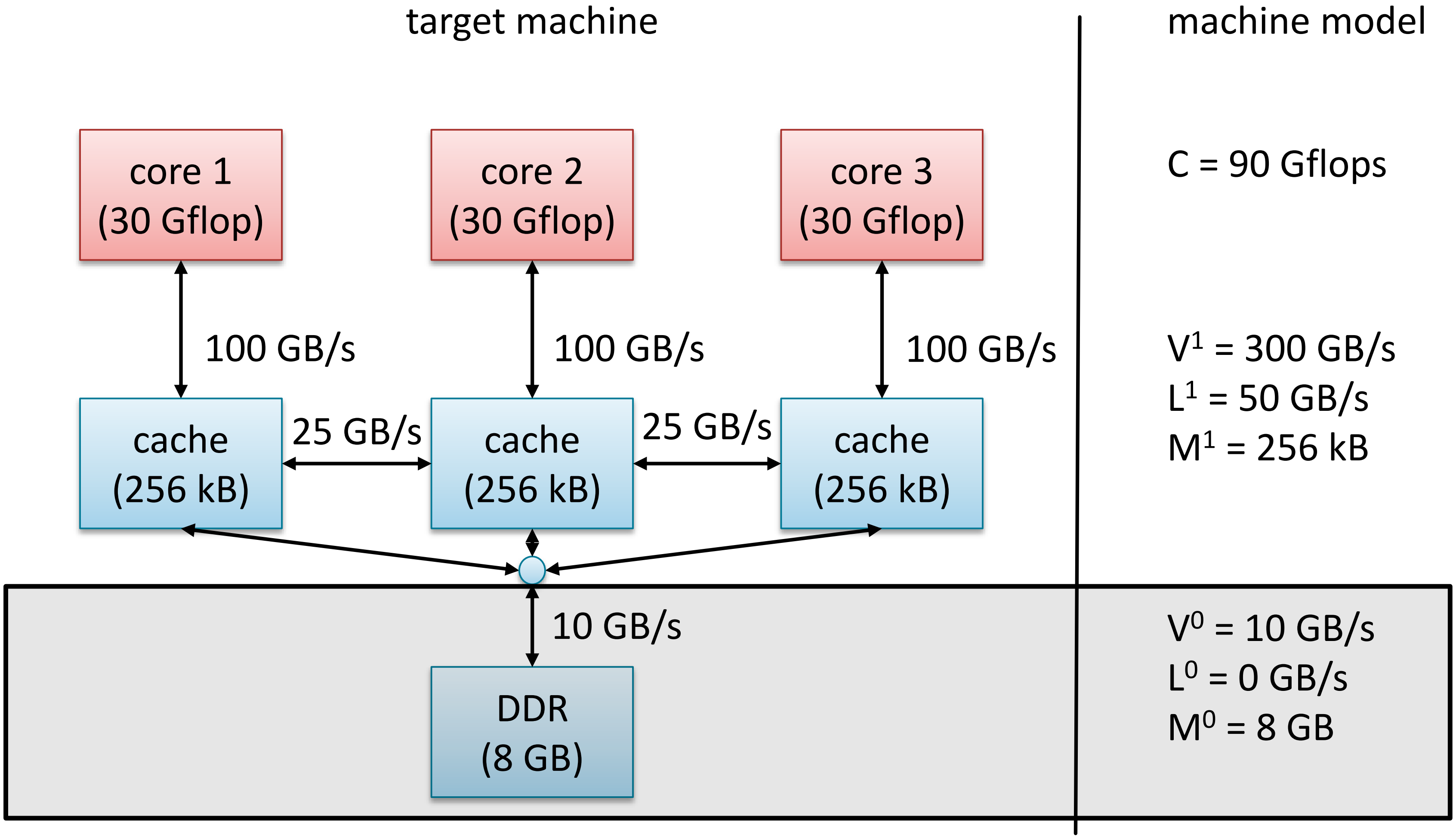
...,lap,fli]],[[flj,out,...

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Machine Performance Mo

lateral and vertical communication refer to communication within one respectively between different tiling hierarchy levels

- **Our model considers peak computation and communication throughputs**



T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

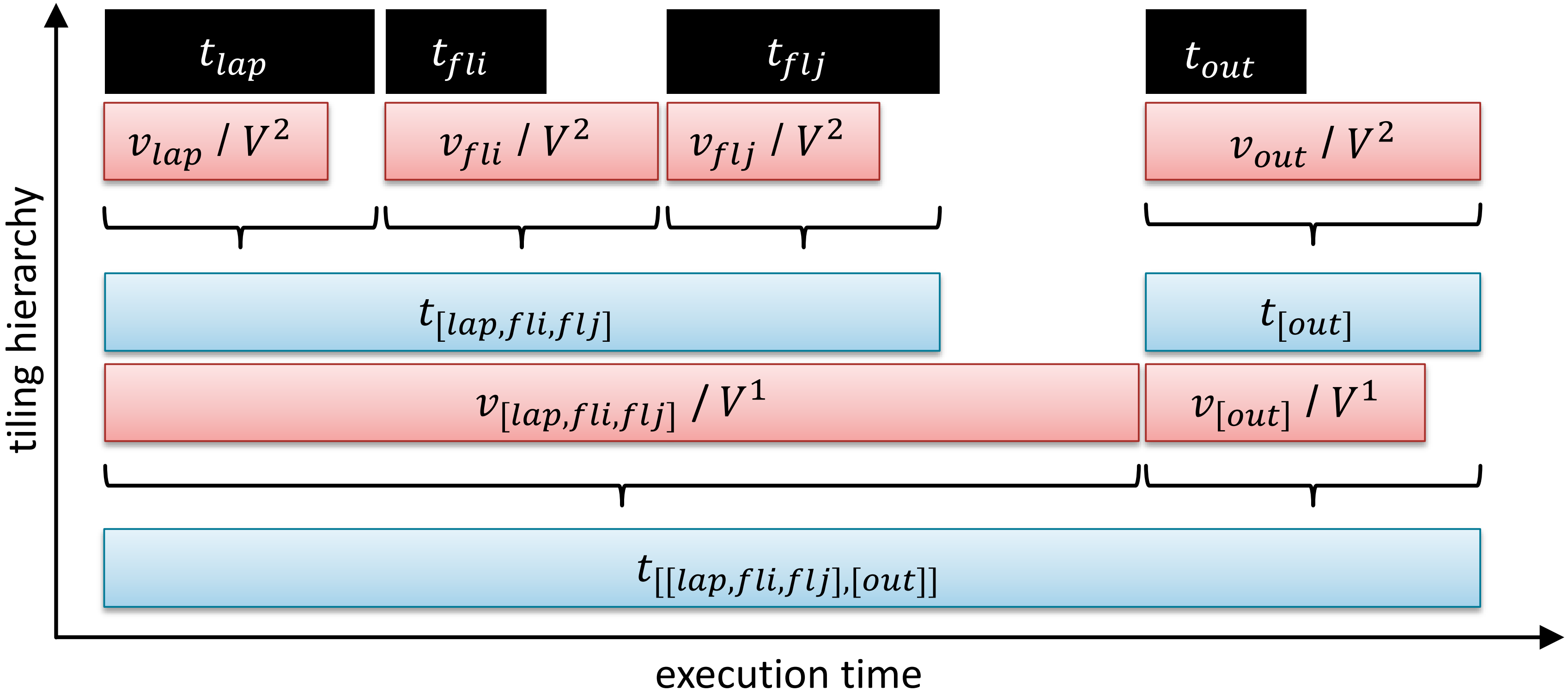
Stencil Performance Model - Overview

- **Given a stencil s given and the amount of computation c_s**

$$t_s = c_s / C$$

- **Given a group g and the vertical and lateral communication v_c and $l_c^1, \ldots, l_c^m$**

$$t_g = \sum_{c \in g.child} \max(t_c, v_c / V^m, l_c^1 / L^1, \ldots, l_c^m / L^m)$$



T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Stencil Performance Model - Affine Sets and Maps

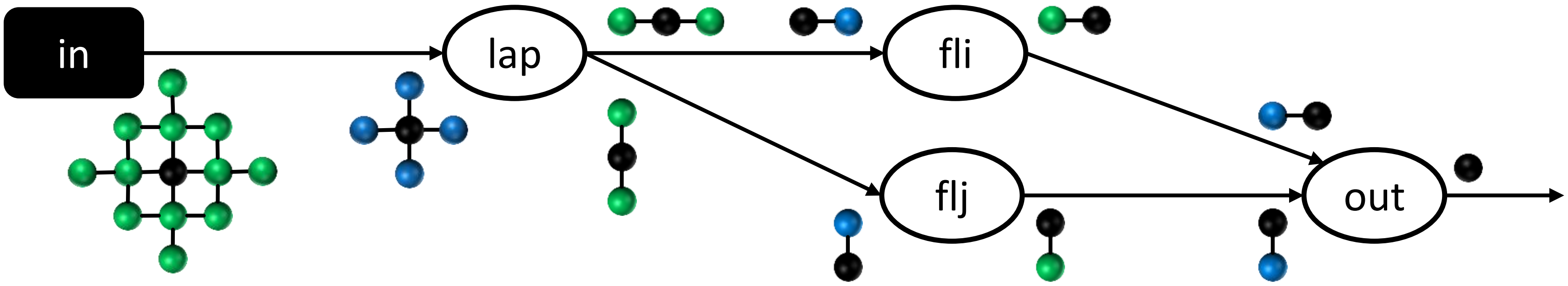
- **The stencil program analysis is based on (quasi-) affine sets and maps**

$$S = \{\vec{i} \mid \vec{i} \in \mathbb{Z}^n \wedge (0, \dots, 0) < \vec{i} < (10, \dots, 10)\}$$

$$M = \{\vec{i} \rightarrow \vec{j} \mid \vec{i} \in \mathbb{Z}^n, \vec{j} \in \mathbb{Z}^n \wedge \vec{j} = 2 \cdot \vec{i}\}$$

- **For example, data dependencies can be expressed using named maps**

$$D_{fli} = \{(fli, \vec{i}) \rightarrow (lap, \vec{i} + \vec{j}) \mid \vec{i} \in \mathbb{Z}^2, \vec{j} \in \{(0,0), (1,0)\}\}$$



$$D = D_{lap} \cup D_{fli} \cup D_{flj} \cup D_{out}$$

$$E = \mathrm{D}^{+}(\{\ (out, \vec{0})\ \})$$

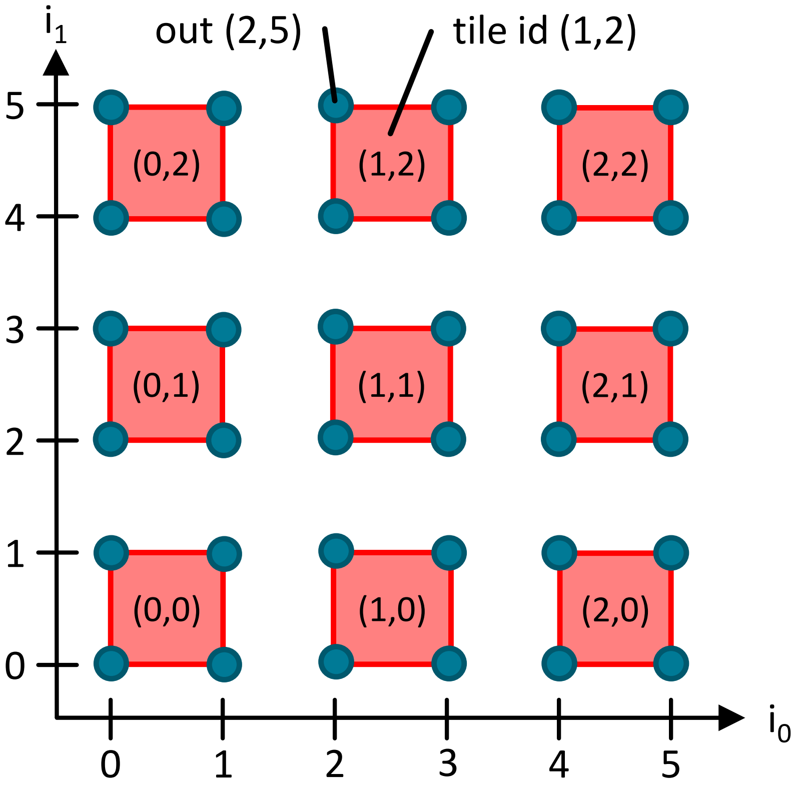
apply the out origin vector to the transitive closure of all dependencies

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS’15

Stencil Performance Model - Tiling Transformations

- **Define a tiling using a map that associates stencil evaluations to tile ids**

$$T_{out} = \{(\text{out}, (i_0, i_1)) \rightarrow (\lfloor i_0/2 \rfloor, \lfloor i_1/2 \rfloor)\}$$



T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

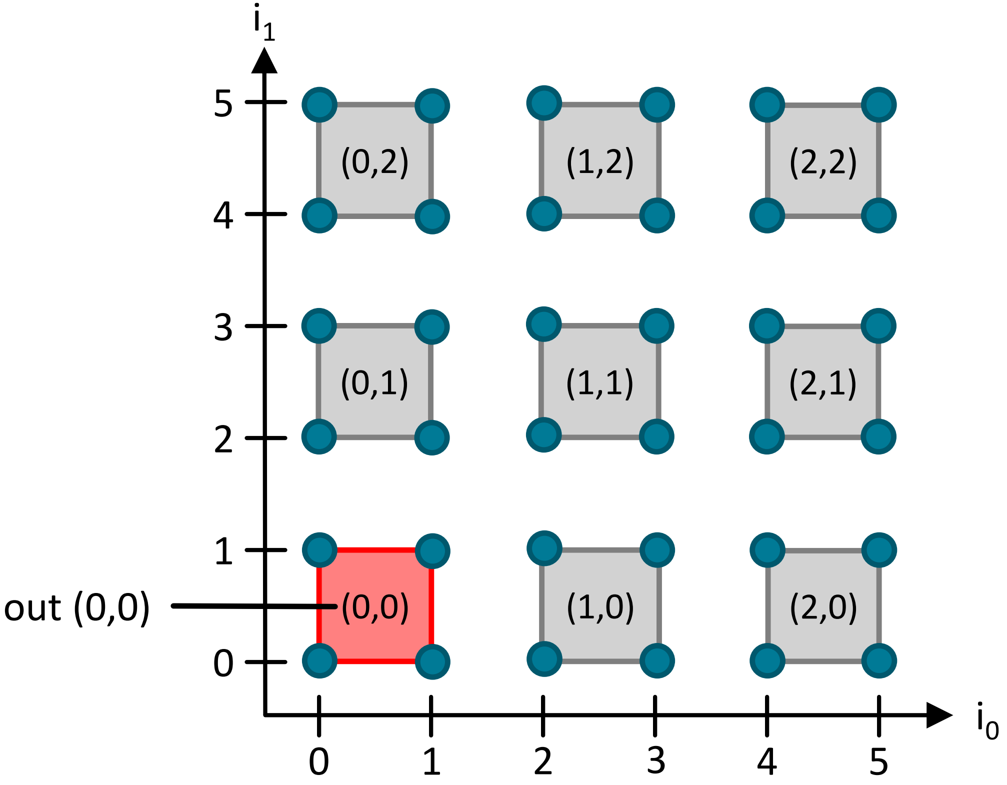
Stencil Performance Model – Comp & Comm

- **Count floating point operations necessary to update tile (0,0)**

$$c_{out} = |T_{out} \cap_{ran} \{(0,0)\}| \cdot \#flops$$

- **Count the number of loads necessary to update tile (0,0)**

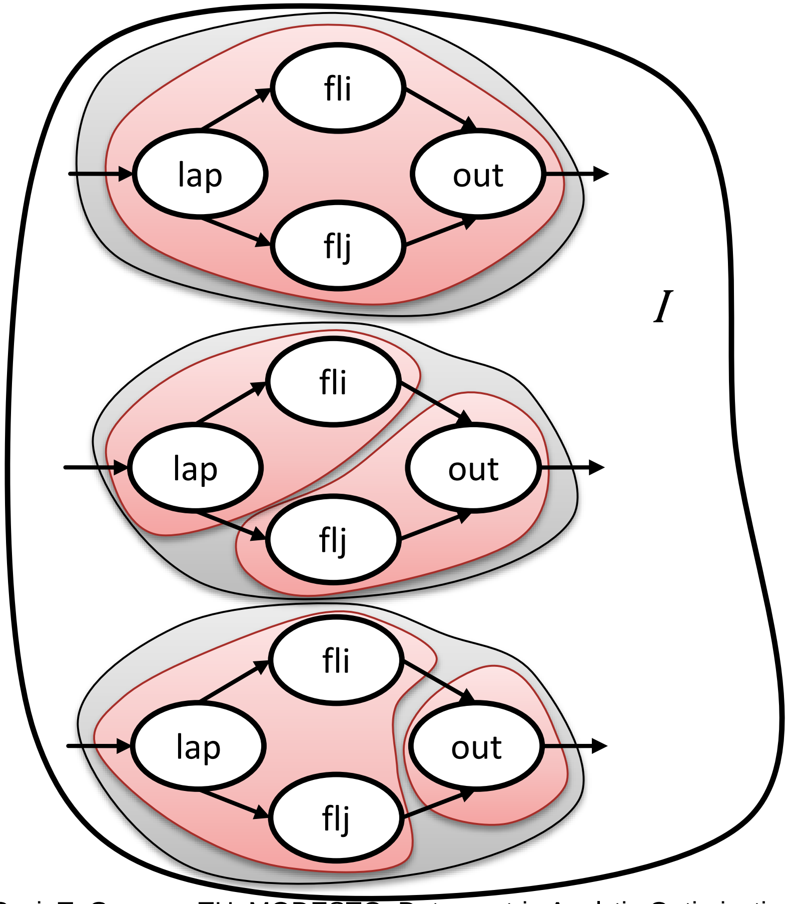
$$l_{out} = |(T_{out} \circ D_{out}^{-1}) \ \cap_{ran} \{(0,0)\}|$$



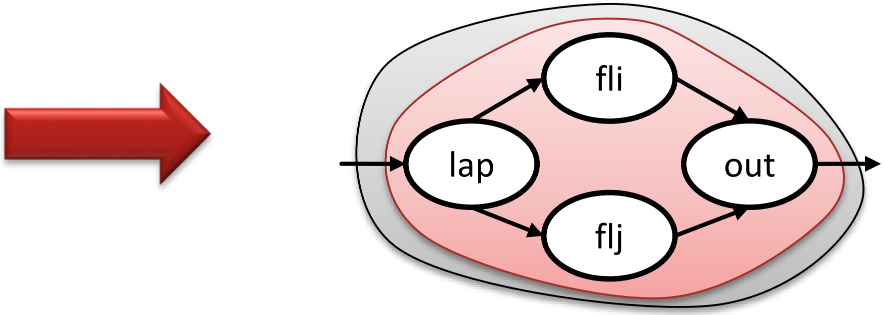
T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Analytic Stencil Program Optimization

- **Put it all together (stencil algebra, performance model, stencil analysis)**
  1. Optimize the stencil execution order (brute force search)
  2. Optimize the stencil grouping (dynamic programming / brute force search)



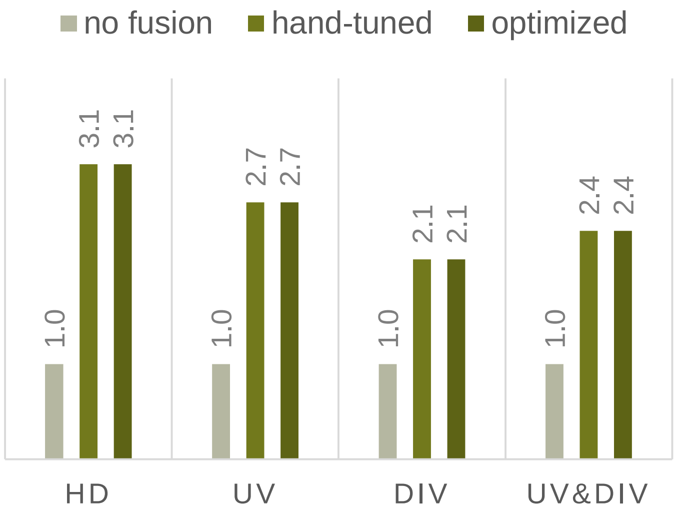
$$\underset{x \in I}{\text{minimize}}\, t(x)$$
$$\text{subject to } m(x) \leq M$$

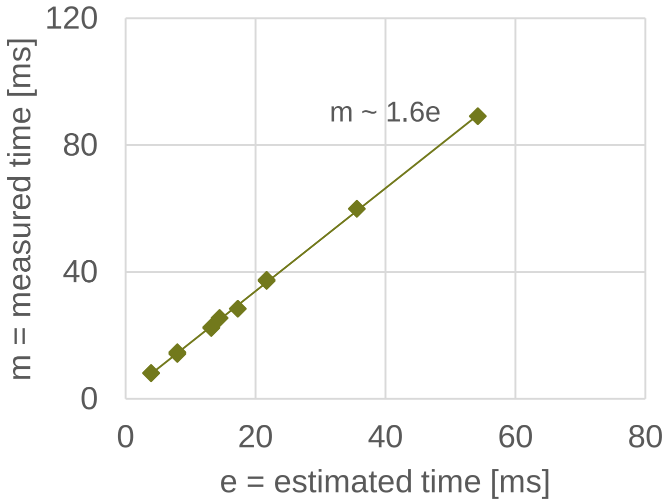


T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS’15

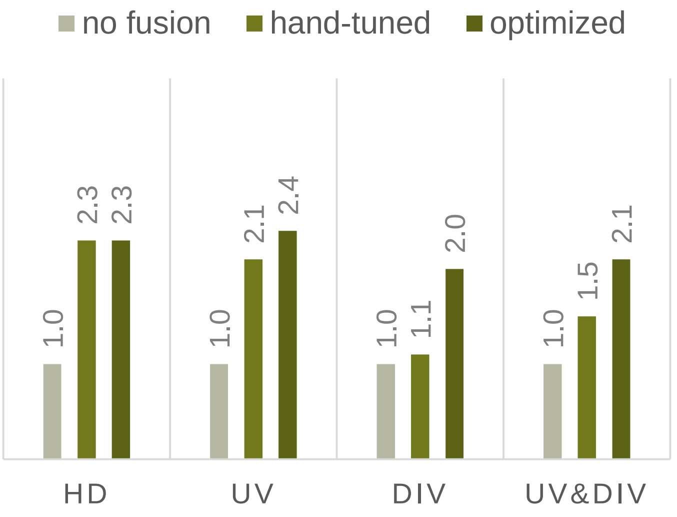
Evaluation

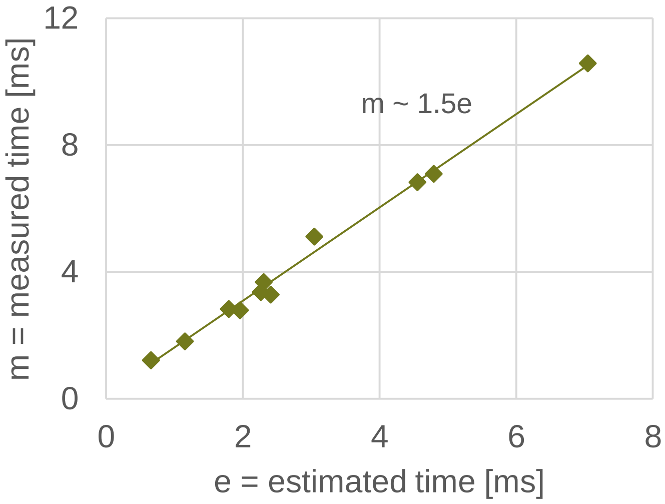
CPU Experiments (i5-3330):





GPU Experiments (Tesla K20c):





T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Not just your basic, average, everyday, ordinary, run-of-the-mill, ho-hum stencil optimizer

- **Complete performance models for:**
  - Computation (very simple)
  - Communication (somewhat tricky, using sets and Minkowski sums, parts of the PM)
- **Established a stencil algebra**
  - Complete enumeration of **all** program variants
- **Navigate the performance space analytically**
  - Find the best program variant for a given system

    Very different for CPU and GPU!
- **Automatic tuning of stencil programs (using the STELLA DS(e)L)**
  - 2.0-3.1x speedup against naive implementations
  - 1.0-1.8x speedup against expert tuned implementations

Sponsors: ETH-RAT   SUK·CUS Schweizerische Universitätskonferenz   PASC17 Platform for Advanced Scientific Computing Conference Lugano Switzerland 26-28 June 2017

T. Gysi, T. Grosser, TH: MODESTO: Data-centric Analytic Optimization of Complex Stencil Programs on Heterogeneous Architectures, ACM ICS'15

Backup Slides