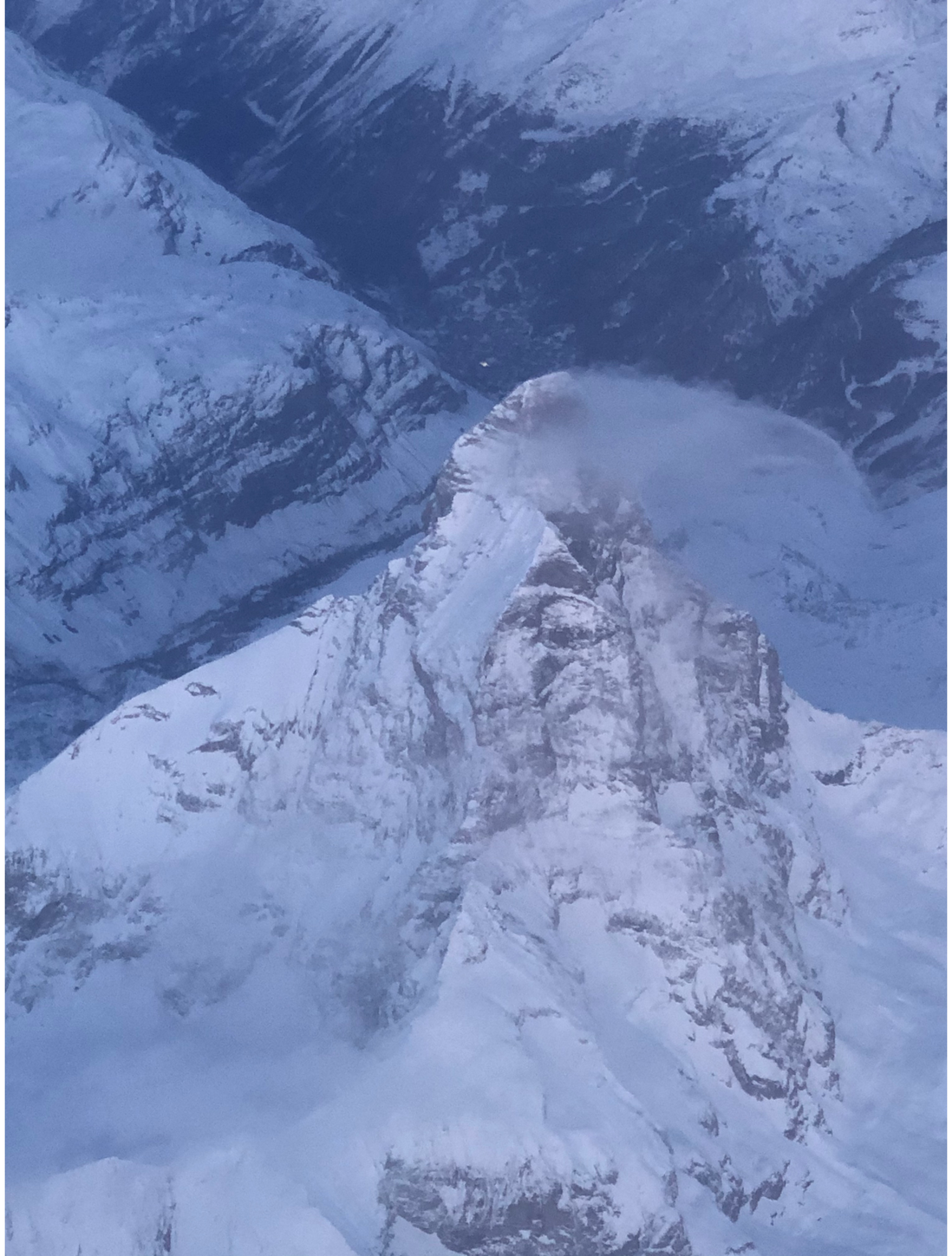
ETH zürich

Bridging the software and performance gap to exascale for weather and climate simulations

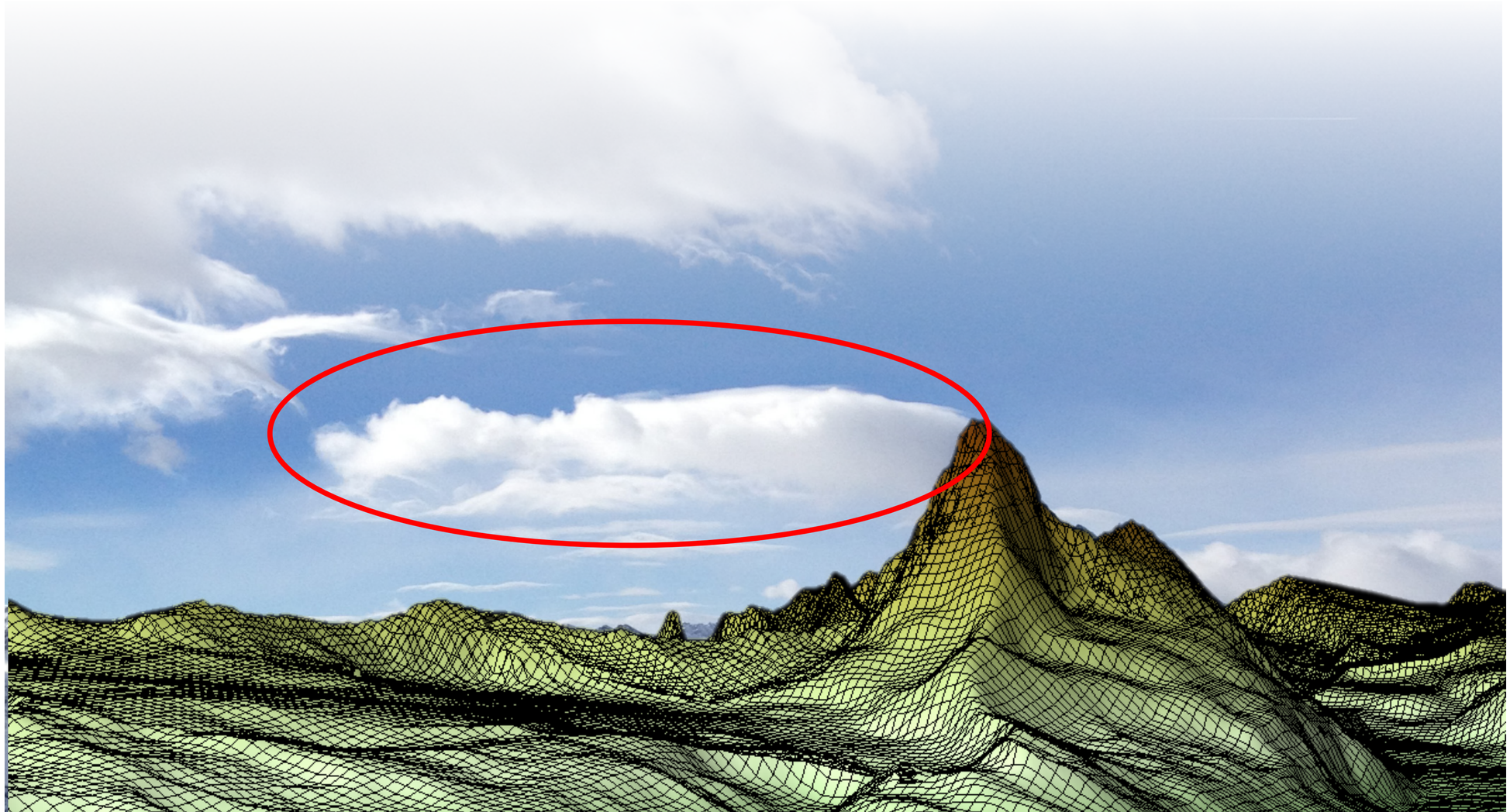
Thomas C. Schulthess

ETH zürich



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Why extreme-scale computing in Switzerland?

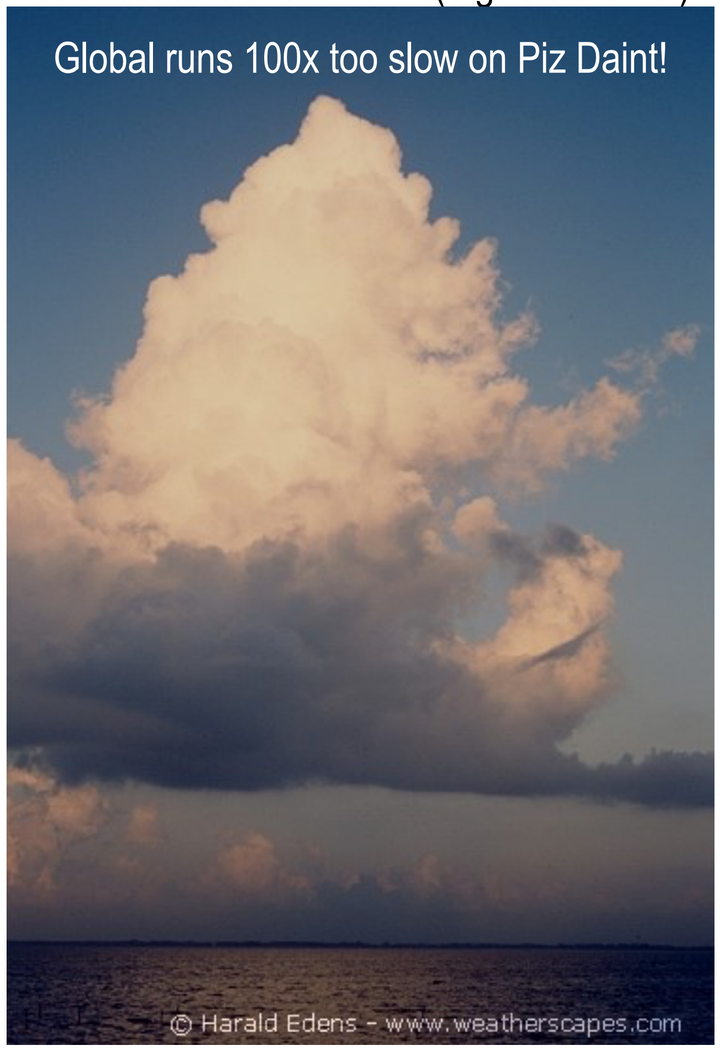


Why extreme-scale computing in Switzerland?

- What computer performance would be necessary to include banner clouds of the Matterhorn in operational weather forecasting?
- A factor 2x in resolution corresponds approximately to a factor 10x in compute performance
- **Δx = 2200 m** (COSMO-2, until 2015)
- Factor x 1’000’000!

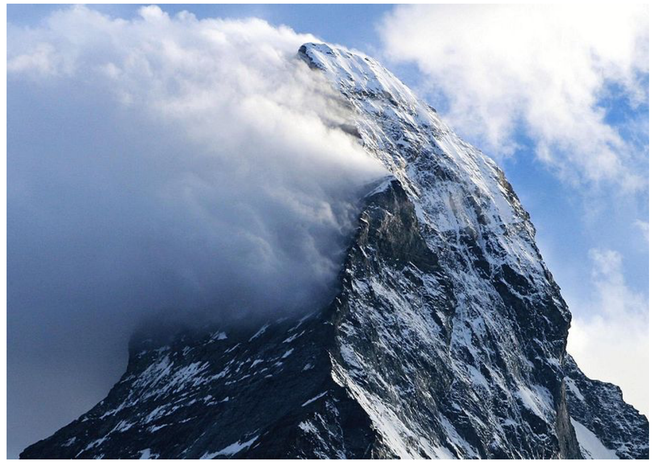
Where we are today, in 2019, and where we go next

Convective clouds ~ 1km (e.g. COSMO-1)



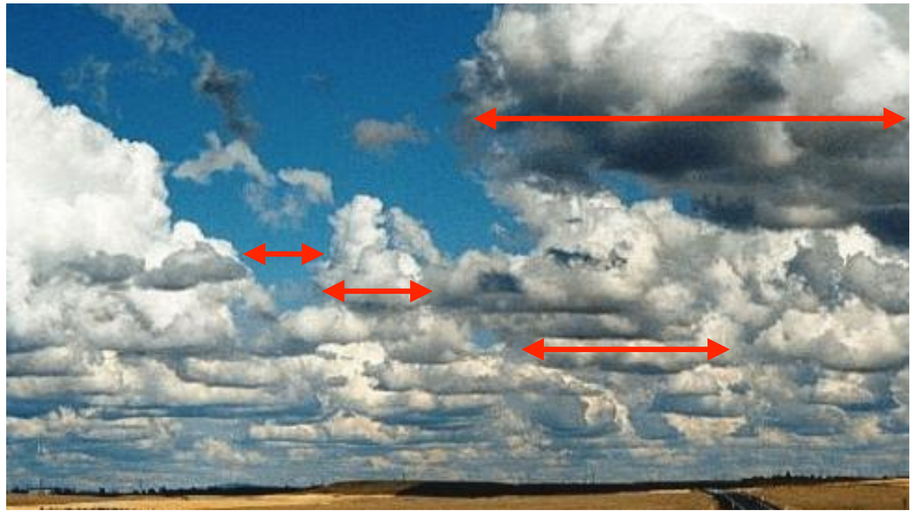
Fog ~ 100 m – will need 1000x more

Banner clouds ~ 10 m – will need 100’000x more



Resolving convective clouds (convergence?)

Structural convergence



Statistics of cloud ensemble:
E.g., spacing and size of convective clouds

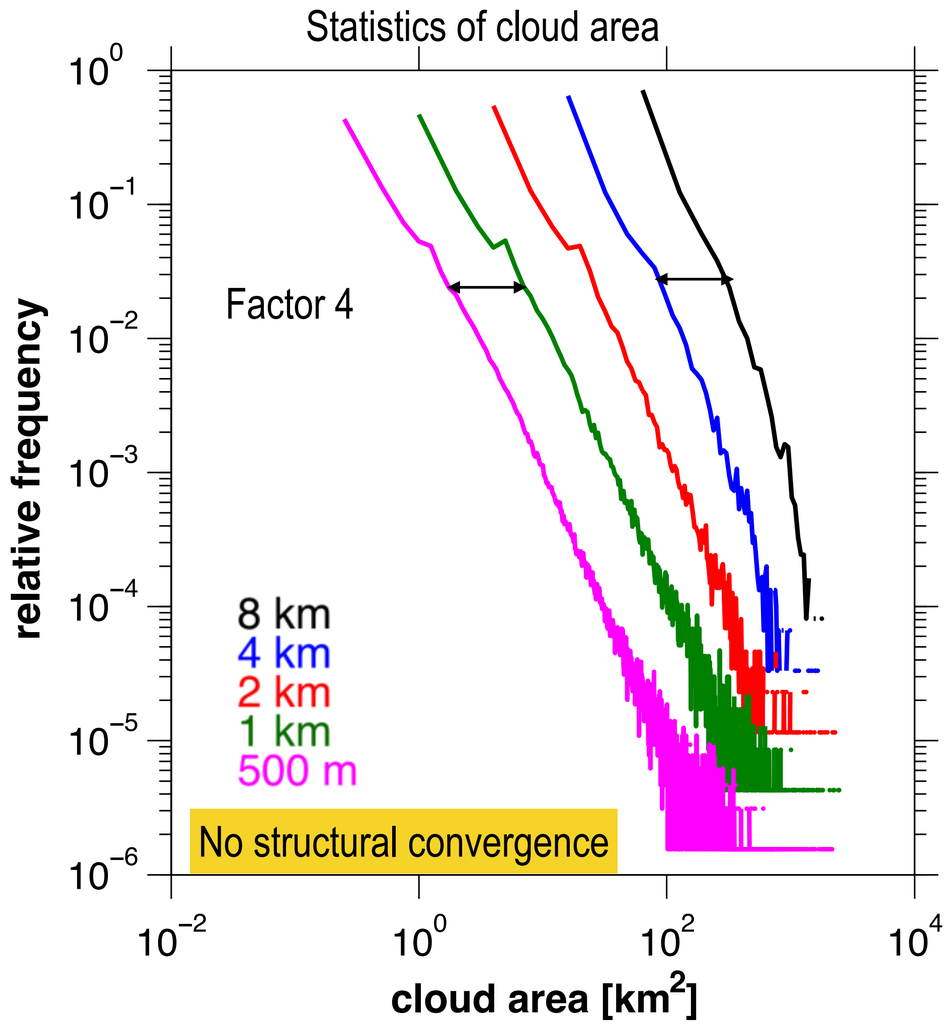
Bulk convergence

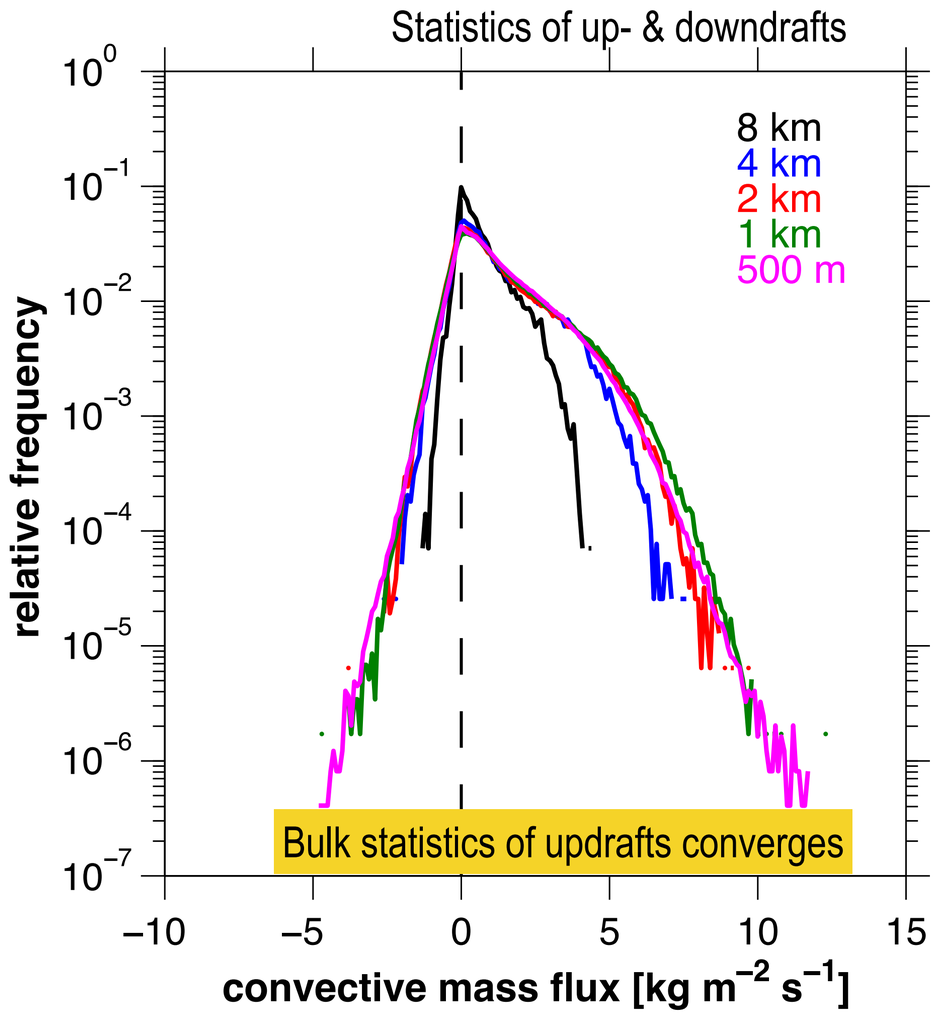
Area-averaged bulk effects upon ambient flow:
E.g., heating and moistening of cloud layer

Source: Christoph Schär, ETH Zurich

Structural and bulk convergence

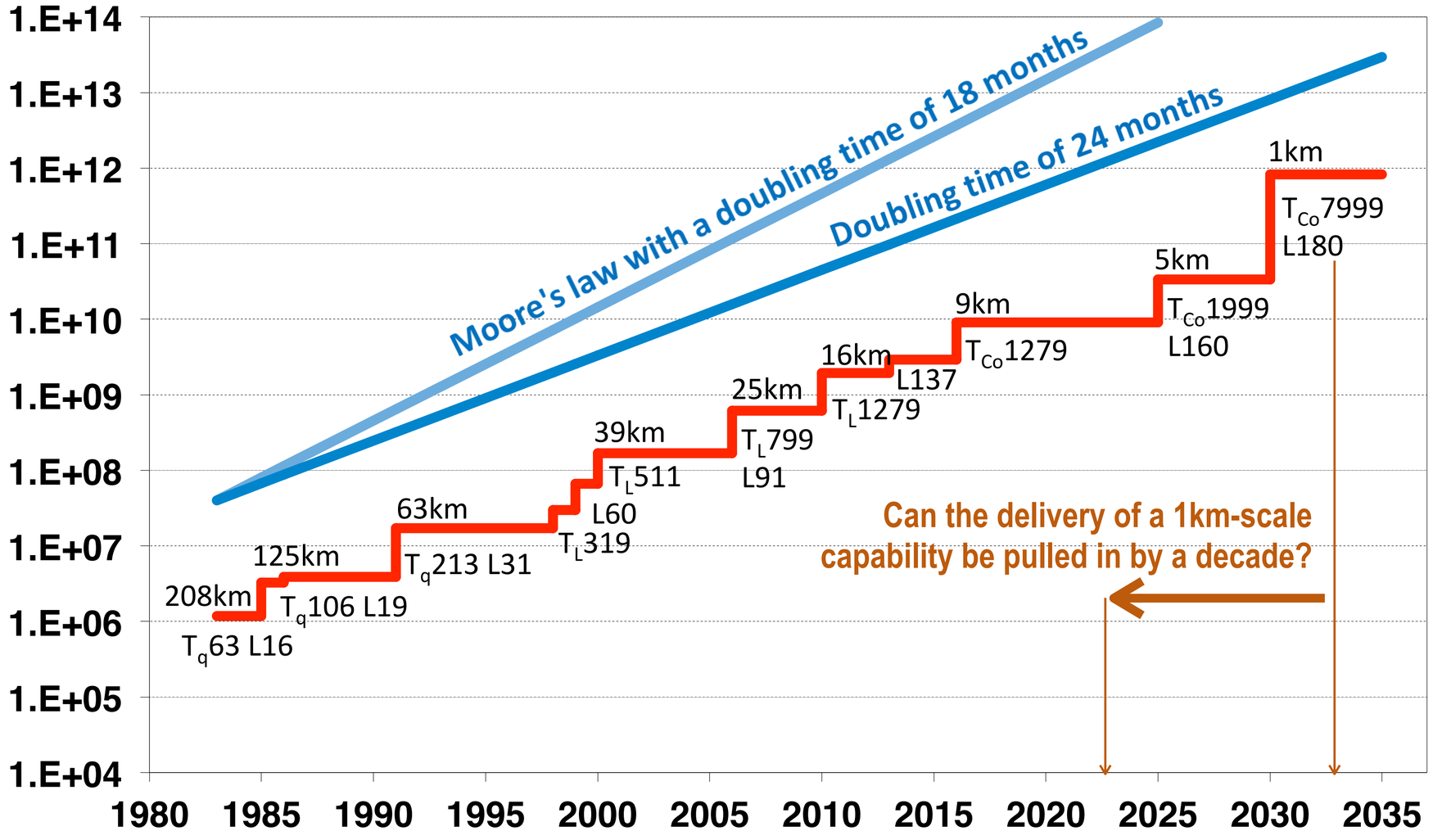
(Panosetti et al. 2018)





Source: Christoph Schär, ETH Zurich

Computational power drives spatial resolution



Source: Christoph Schär, ETH Zurich, & Nils Wedi, ECMWF

Our “exascale” goal for 2022

Horizontal resolution	1 km (globally quasi-uniform)
Vertical resolution	180 levels (surface to ~100 km)
Time resolution	Less than 1 minute
Coupled	Land-surface/ocean/ocean-waves/sea-ice
Atmosphere	Non-hydrostatic
Precision	Single (32bit) or mixed precision
Compute rate	1 SYPD (simulated year wall-clock day)

Running COSMO 5.0 & IFS (“the European Model”) at global scale on Piz Daint

Scaling to full system size: ~5300 GPU accelerate nodes available

Running a near-global (±80º covering 97% of Earths surface) COSMO 5.0 simulation & IFS

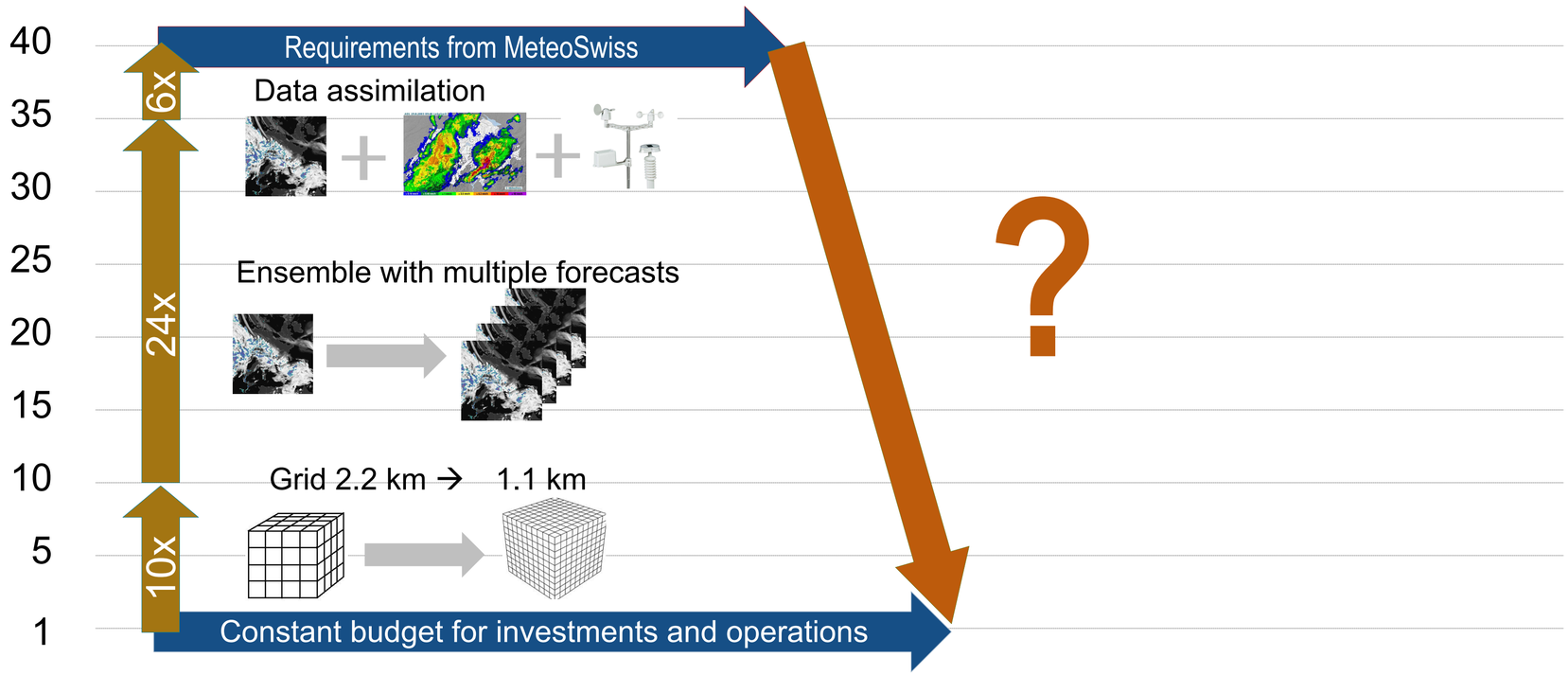
> Either on the hosts processors: Intel Xeon E5 2690v3 (Haswell 12c).

> Or on the GPU accelerator: PCIe version of NVIDIA GP100 (Pascal) GPU

The baseline for COSMO-global and IFS

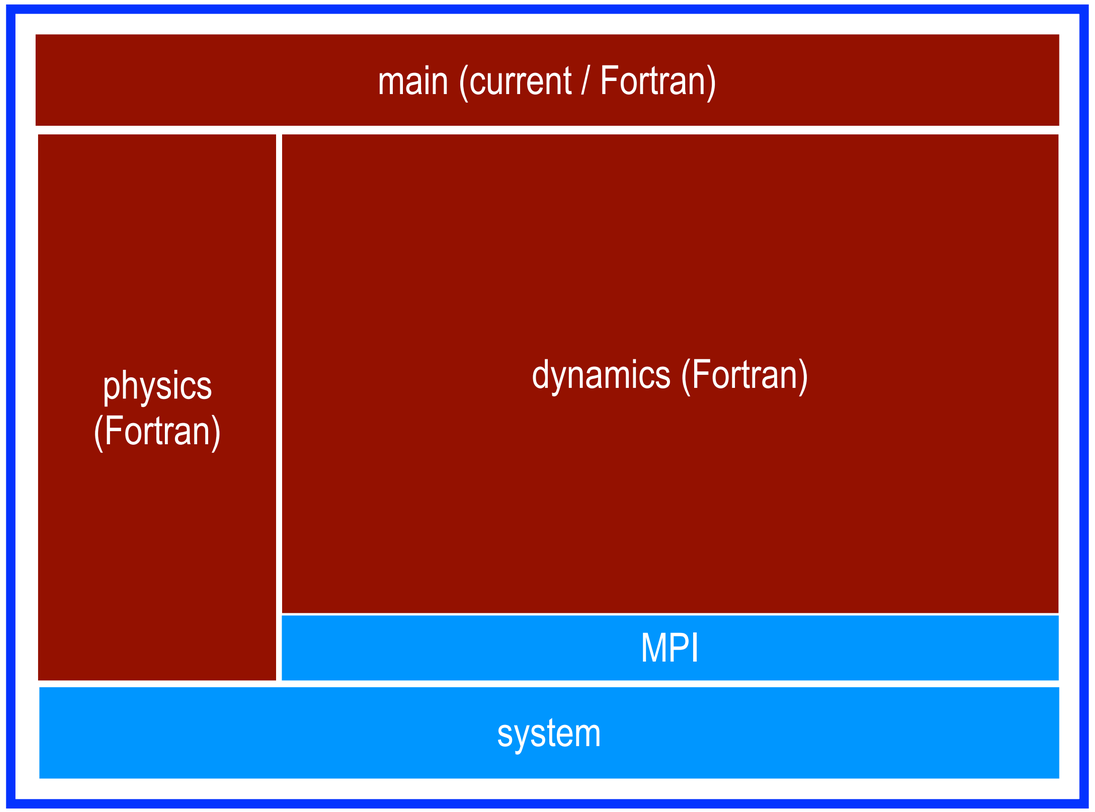
	Near-global COSMO[15]		Global IFS[16]	
	Value	Shortfall	Value	Shortfall
Horizontal resolution	0.93 km (non-uniform)	0.81×	1.25 km	1.56×
Vertical resolution	60 levels (surface to 25 km)	3×	62 levels (surface to 40 km)	3×
Time resolution	6 s (split-explicit with sub-stepping)*	–	120 s (semi-implicit)	4×
Coupled	No	1.2×	No	1.2×
Atmosphere	Non-hydrostatic	–	Non-hydrostatic	–
Precision	Single	–	Single	–
Compute rate	0.043 SYPD	23×	0.088 SYPD	11×
Other (e.g., physics, . . .)	microphysics	1.5×	Full physics	–
Total shortfall		101×		247×

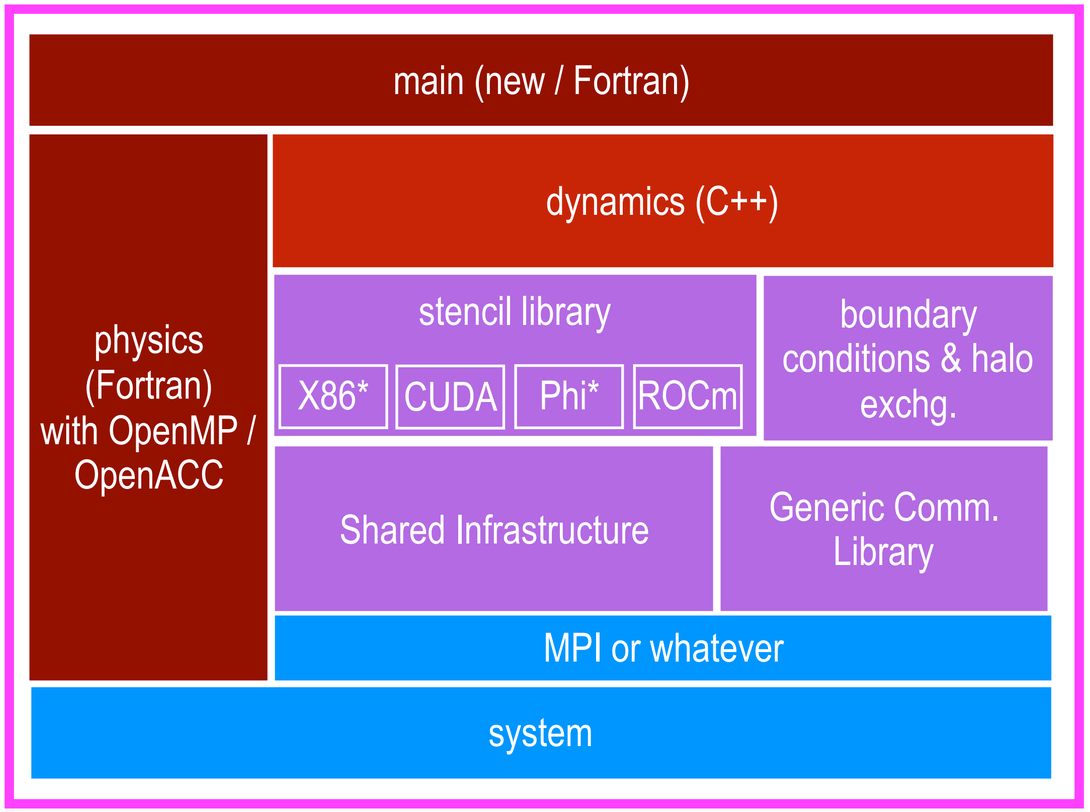
MeteoSwiss' performance ambitions in 2013



We need a 40x improvement between 2012 and 2015 at constant cost

COSMO: old and new (refactored) code

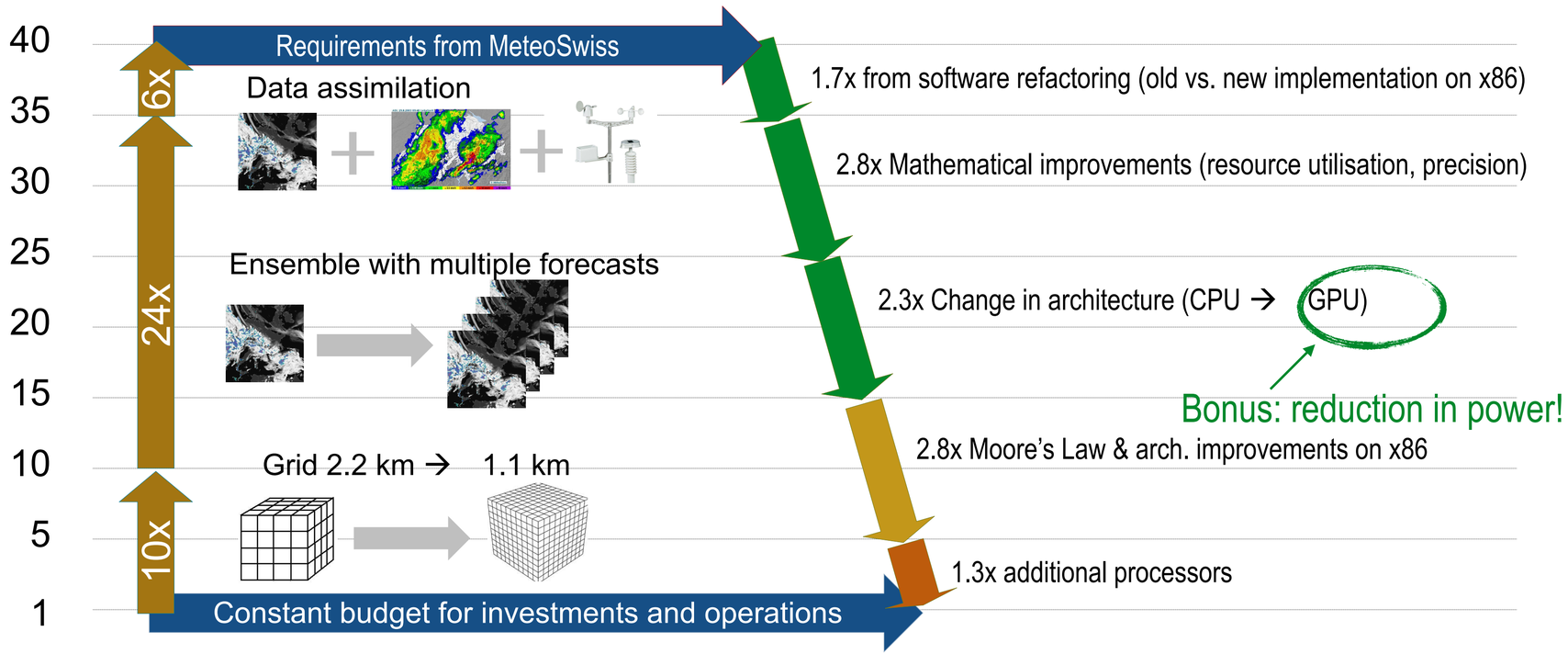




* two different OpenMP backends

Where the factor 40 improvement came from

Investment in software allowed mathematical improvements and change in architecture



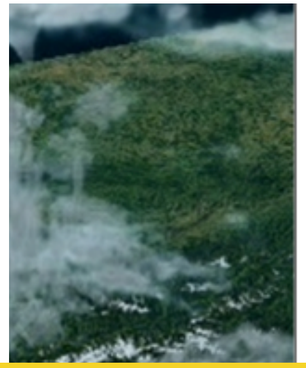
There is no silver bullet!

HPC wire

September 15, 2015

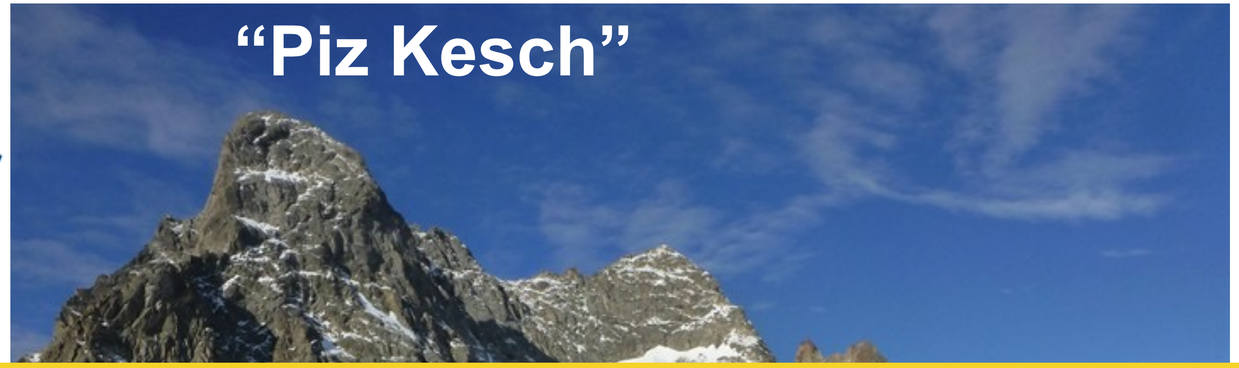
Today's Outlook: GPU-accelerated Weather Forecasting

John Russell



MeteoSwiss New Weather Supercomputer

World's First GPU-Accelerated Weather Forecasting System



Since April 2016, the Swiss version* of the COSMO model is running operationally on GPUs

2x Racks

48 CPUs

192 Tesla K80 GPUs

> 90% of FLOPS from GPUs

Operational in 2016

5 NVIDIA

(*) Swiss version of the COSMO model is running at 1km horizontal resolution over Alpine region and was (in 2016) ~10x more efficient than the state of the art

The baseline for COSMO-global and IFS

	Near-global COSMO[15]		Global IFS[16]	
	Value	Shortfall	Value	Shortfall
Horizontal resolution	0.93 km (non-uniform)	0.81×	1.25 km	1.56×
Vertical resolution	60 levels (surface to 25 km)	3×	62 levels (surface to 40 km)	3×
Time resolution	6 s (split-explicit with sub-stepping)*	–	120 s (semi-implicit)	4×
Coupled	No			1.2×
Atmosphere	Non-hydrostatic	–	Non-hydrostatic	–
Precision	Single	–	Single	–
Compute rate	0.043 SY	3×	0.088 SYPD	11×
Other (e.g., physics, . . .)	microphysics	1.5×	Full physics	–
Total shortfall		101×		247×

100x (single trajectory) times 50x (ensemble)

Goal is to stay within ~ 5MW

Memory use efficiency

Fuhrer et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-230, published 2018

Necessary data transfers

Achieved BW

$$MUE = \text{I/O efficiency} \cdot \text{BW efficiency} = \frac{Q}{D}\frac{B}{\hat{B}}$$

0.88

= 0.67

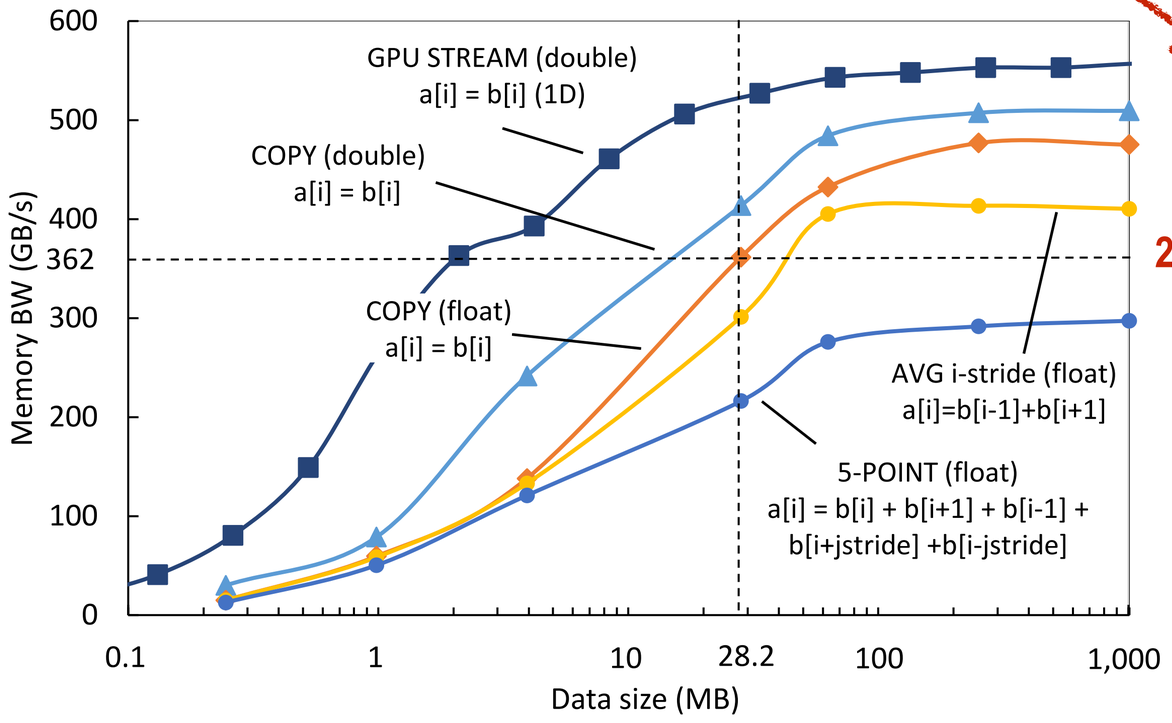
Actual data transfers

0.76

Max achievable BW (STREAM)

0.55 w. regard to peak BW

2x lower than peak BW



Can the 100x shortfall of a grid-based implementation like COSMO-global be overcome?

1. Icosahedral/octahedral grid (ICON/IFS) vs. Lat-long/Cartesian grid (COSMO)

   2x fewer grid-columns

   Time step of 10 ms instead of 5 ms

   4x

2. Improving BW efficiency

   Improve BW efficiency and peak BW

   (results on Volta show this is realistic)

   2x

3. Strong scaling

   4x possible in COSMO, but we reduced available parallelism by factor 1.33

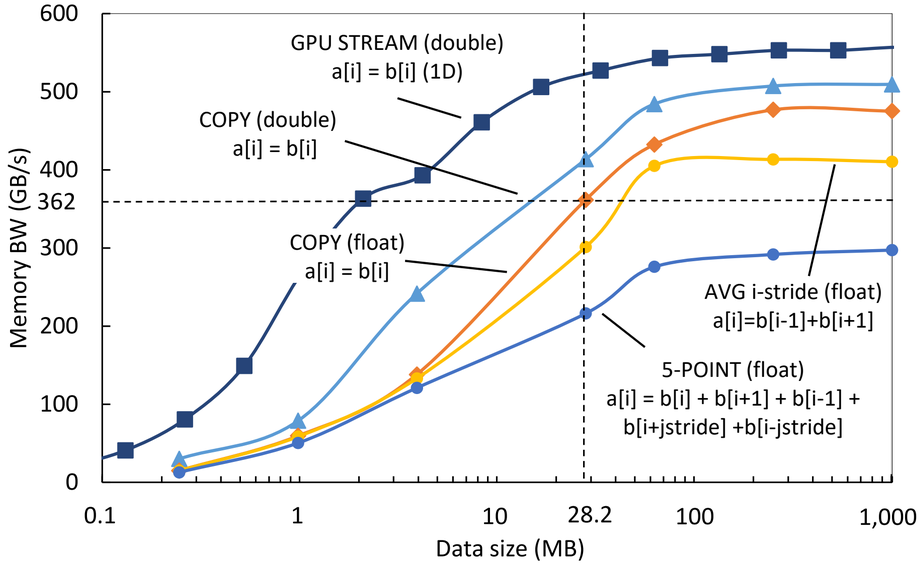
   3x

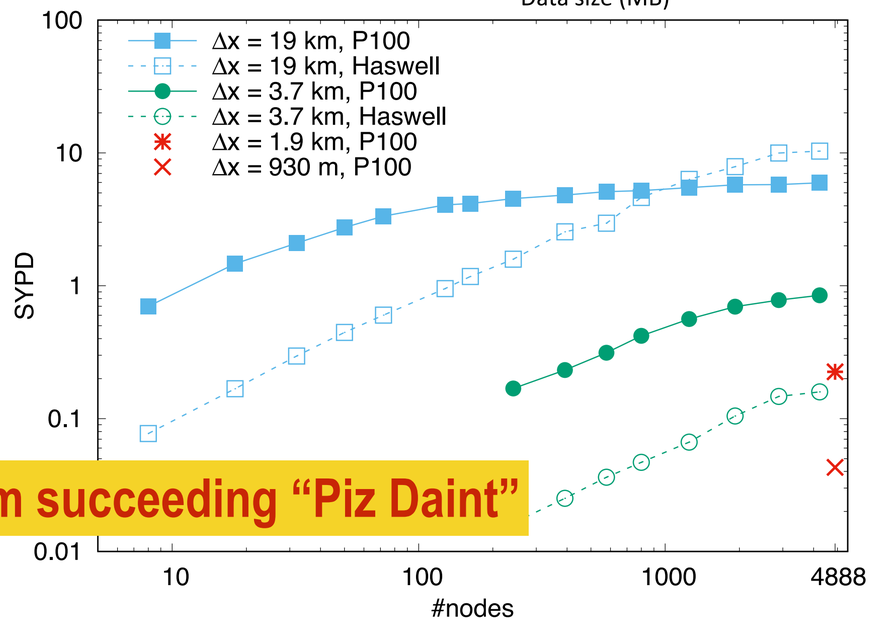
4. Remaining reduction in shortfall

   Numerical algorithms (larger time steps)

   Further improved processors / memory

   4x





But we don't want to increase the footprint of the 2022 system succeeding "Piz Daint"

Much of the data present here was from this article

Race to Exascale Computing

Theme Article

Reflecting on the Goal and Baseline for Exascale Computing: A Roadmap Based on Weather and Climate Simulations

Thomas C. Schulthess
ETH Zurich, Swiss National Supercomputing Centre

Peter Bauer
European Centre for Medium-Range Weather Forecasts

Nils Wedi
European Centre for Medium-Range Weather Forecasts

Oliver Fuhrer
MeteoSwiss

Torsten Hoefler
ETH Zurich

Christoph Schär
ETH Zurich

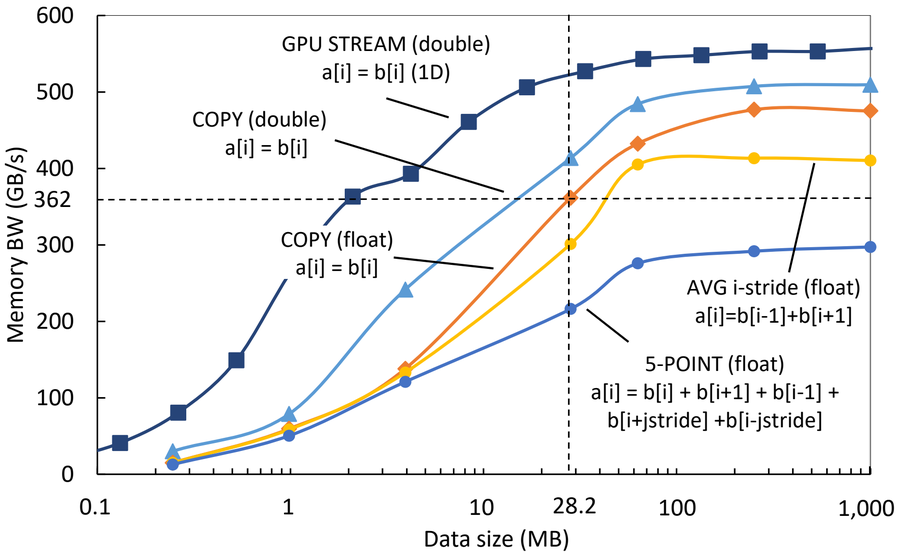
***Abstract*—We present a roadmap towards exascale computing based on true application performance goals. It is based on two state-of-the art European numerical weather prediction models (IFS from ECMWF and COSMO from MeteoSwiss) and their current performance when run at very high spatial resolution on present-day supercomputers. We conclude that these models execute about 100–250 times too slow for operational throughput rates at a horizontal resolution of 1 km, even when executed on a full petascale system with nearly 5000 state-of-the-art hybrid GPU-CPU nodes. Our analysis of the performance in terms of a metric that assesses the efficiency of memory use shows a path to improve the performance of hardware and software in order to meet operational requirements early next decade.**

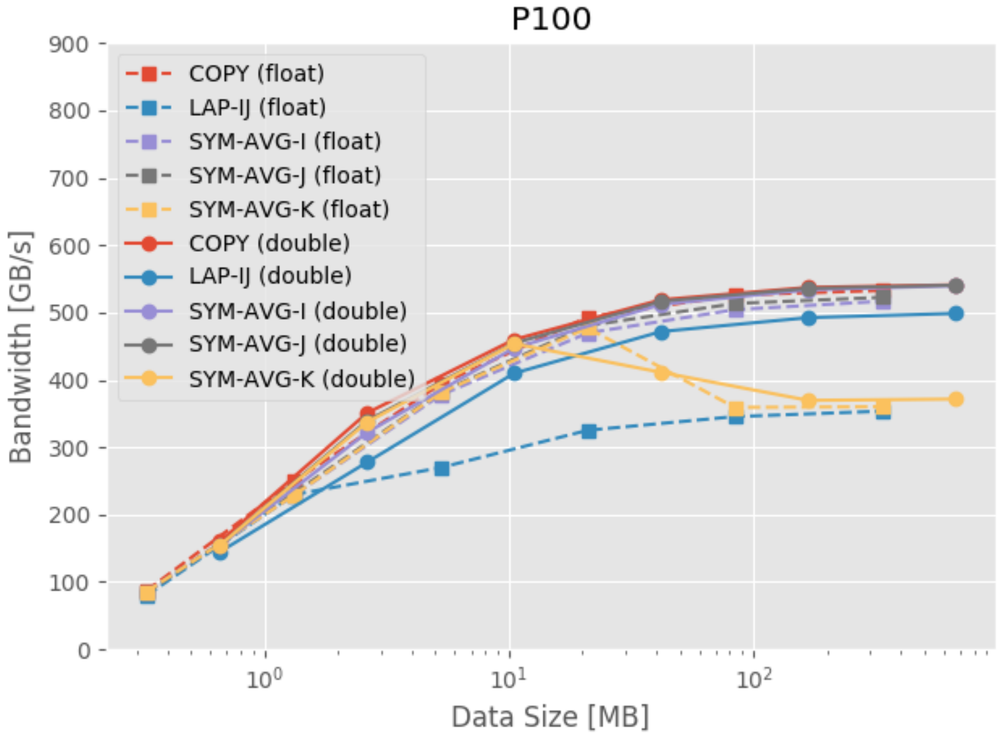
Digital Object Identifier 10.1109/MCSE.2018.2888788
Date of publication 24 December 2018; date of current version 6 March 2019.

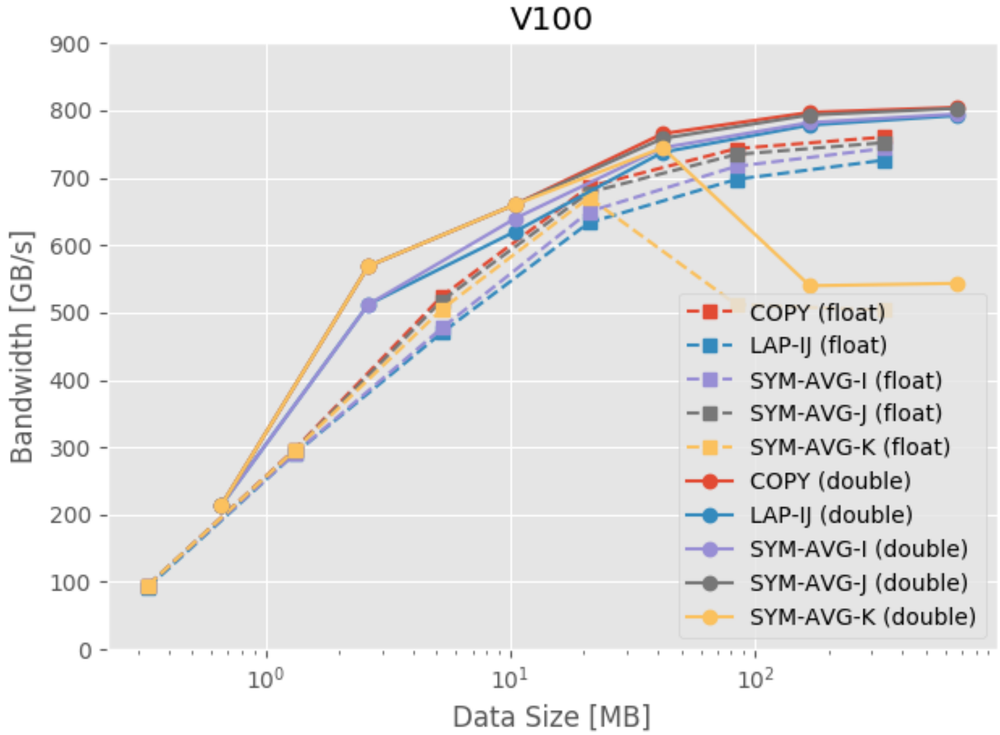
Scientific computation with precise numbers has always been hard work, ever since Johannes Kepler analyzed Tycho Brahe's data to

1521-9615 © 2018 IEEE Published by the IEEE Computer Society **Computing in Science & Engineering**

The good news:
memory performance is improving!







What about ensembles and throughput for climate? (Remaining goals beyond 2022)

1. Improve the throughput to 5 SYPD

Change the architecture from control flow to data flow centric (reduce necessary data transfers)

Necessary data transfers

Achieved BW

$$MUE = \text{I/O efficiency} \cdot \text{BW efficiency} = \frac{Q}{D}\frac{B}{\hat{B}}$$

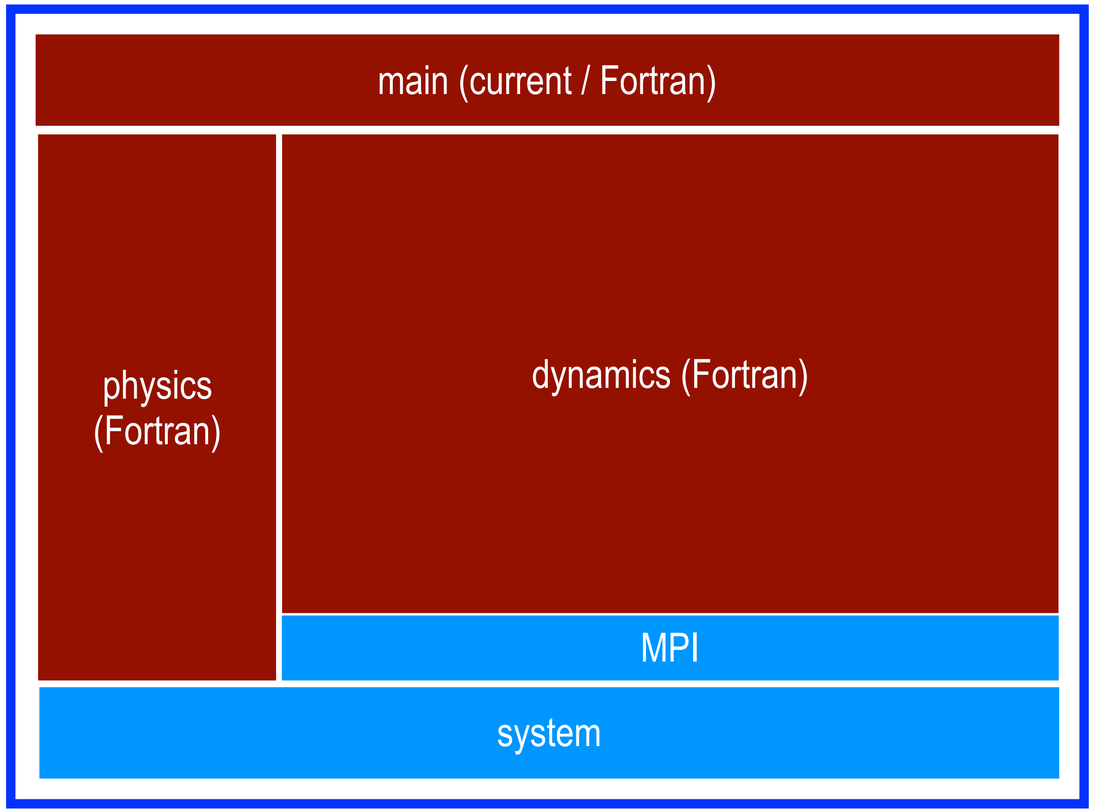
Actual data transfers

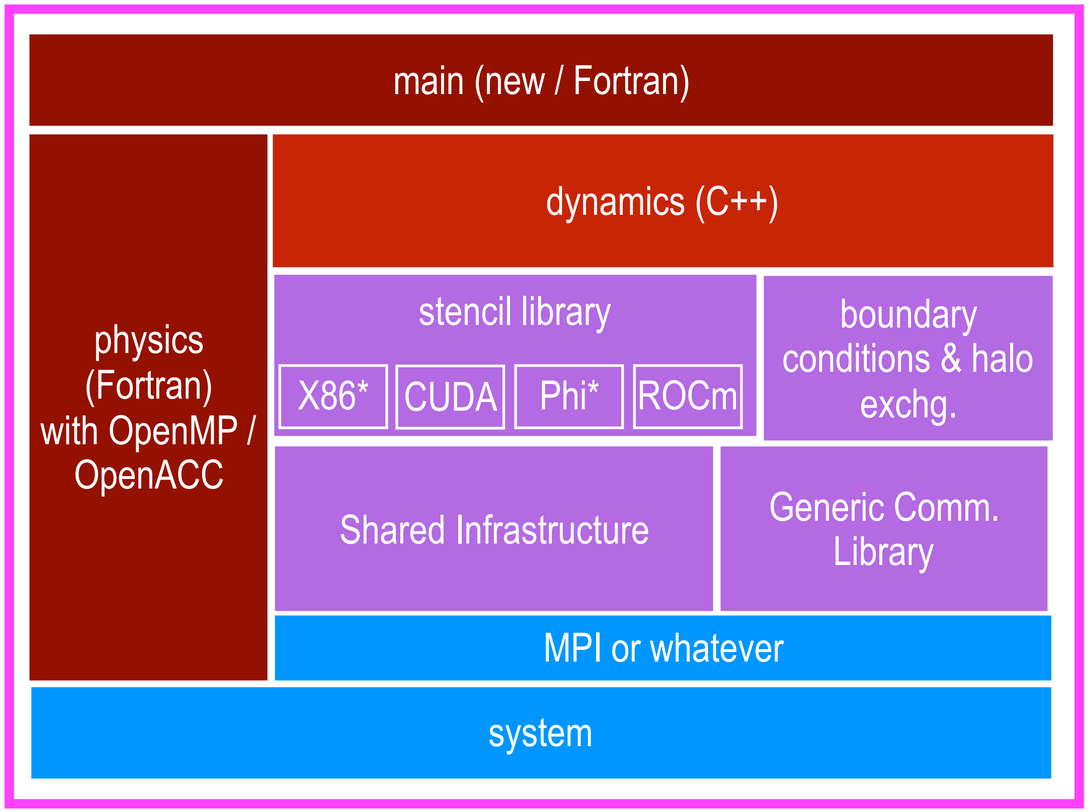
Max achievable BW

2. Reduce the footprint of a single simulation by up to factor 10-50

We may have to change the footprint of machines to hyper scale!

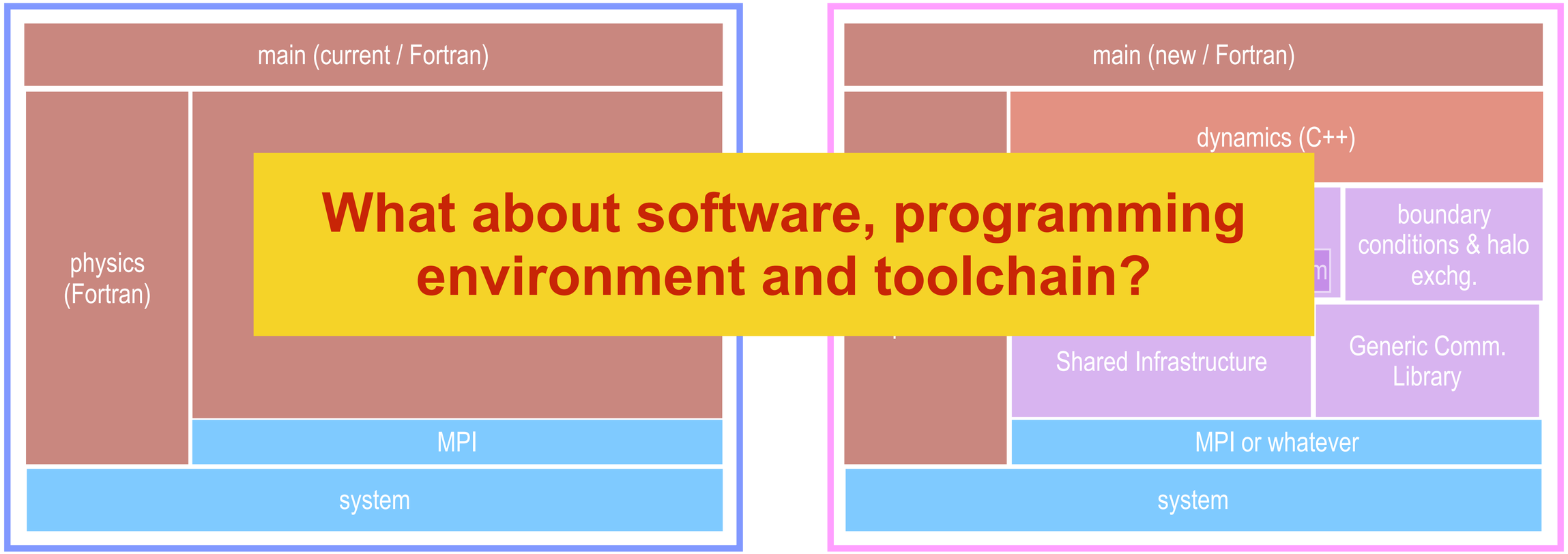
COSMO: old and new (refactored) code





* two different OpenMP backends

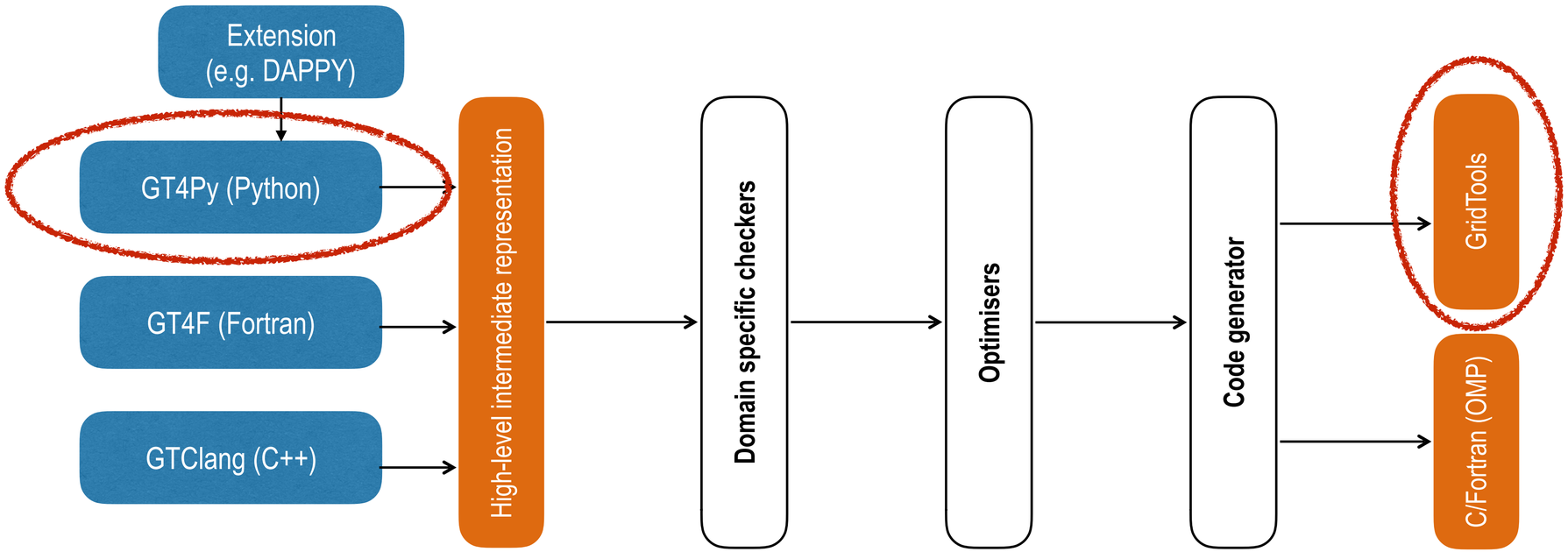
COSMO: old and new (refactored) code



What about software, programming environment and toolchain?

* two different OpenMP backends

Vision for the toolchain

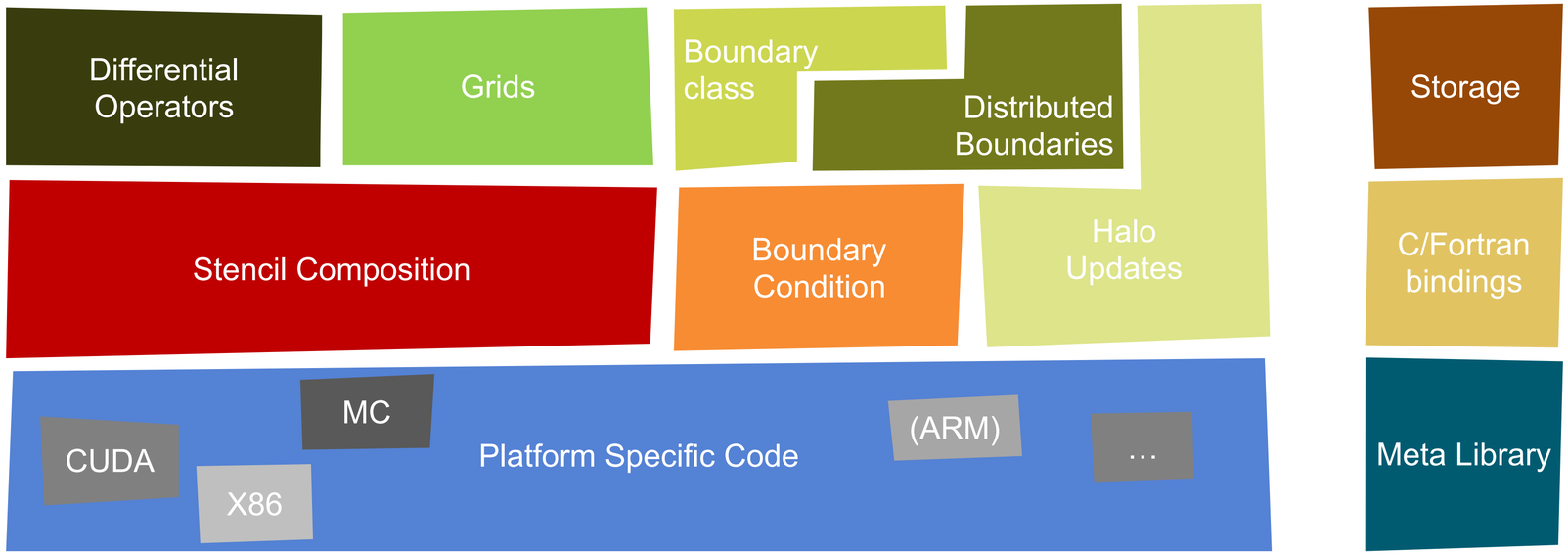


- read before write
- missing boundary update
- data dependency race conditions
- out of bounds stencil access

- Software managed caches
- Full vertical parallelisation
- Stage fusion
- Data locality exploit
- Strong/weak scaling optimiser

- Native C/C++/Fortan
- Optimised GridTools Generator (C++)

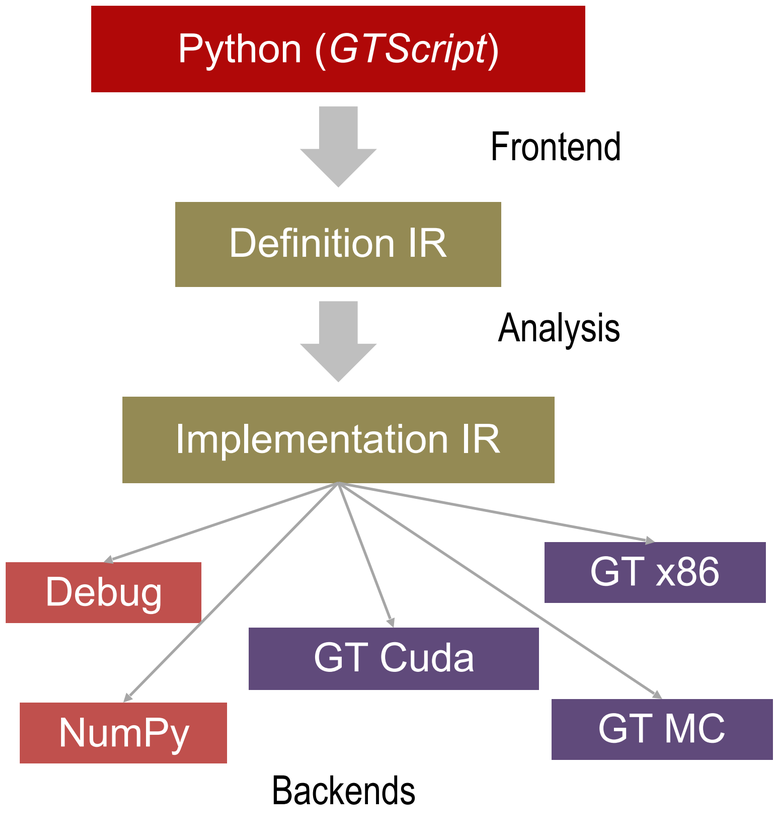
GridTools – released 04/2019, finally!



(Current plan is to use GridTools as performance IR in the 2022 timeframe)

Grid Tools for Python (GT4Py) – arch. of 10/2019 release

- Transformation from Python AST to executable code
- Intermediate representation
  - Definition IR (high-level)
  - Implementation IR (low-level)
- Analysis phase to lower the IR
- Code generation with different targets
  - Python (Debug, NumPy)
  - C++ (GT with X86, Cuda, Phi)



This will work for any other domain
and much better than writing “performance portable”
Fortran/C/C++ code with OpenMP directives!!!

Collaborators on Exascale (climate)


Tim Palmer (U. of Oxford)

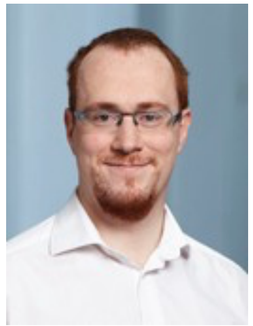

Bjorn Stevens (MPI-M)


Peter Bauer (ECMWF)


Oliver Fuhrer (Vucan)


Nils Wedi (ECMWF)


Carlos Osuna (MeteoSwiss)


Torsten Hoefler (ETH Zurich)


Mauro Bianco (CSCS)


Christoph Schar (ETH Zurich)

As well as the team from CSCS' Scientific Software and Libraries (SLL) section

Thank you!