ETH zürich

Co-Designing GPU-Based Systems for Numerical Weather Predictions

Oliver Fuhrer, André Walser, Colin McMurtrie, Sadaf Alam, Michele De Lorenzi, and Thomas C. Schulthess

ETH zürich

HPC wire

September 15, 2015

Today's Outlook: GPU-accelerated Weather Forecasting

John Russell

"Piz Kesch"

MeteoSwiss New Weather Supercomputer

World's First GPU-Accelerated Weather Forecasting System

2x Racks

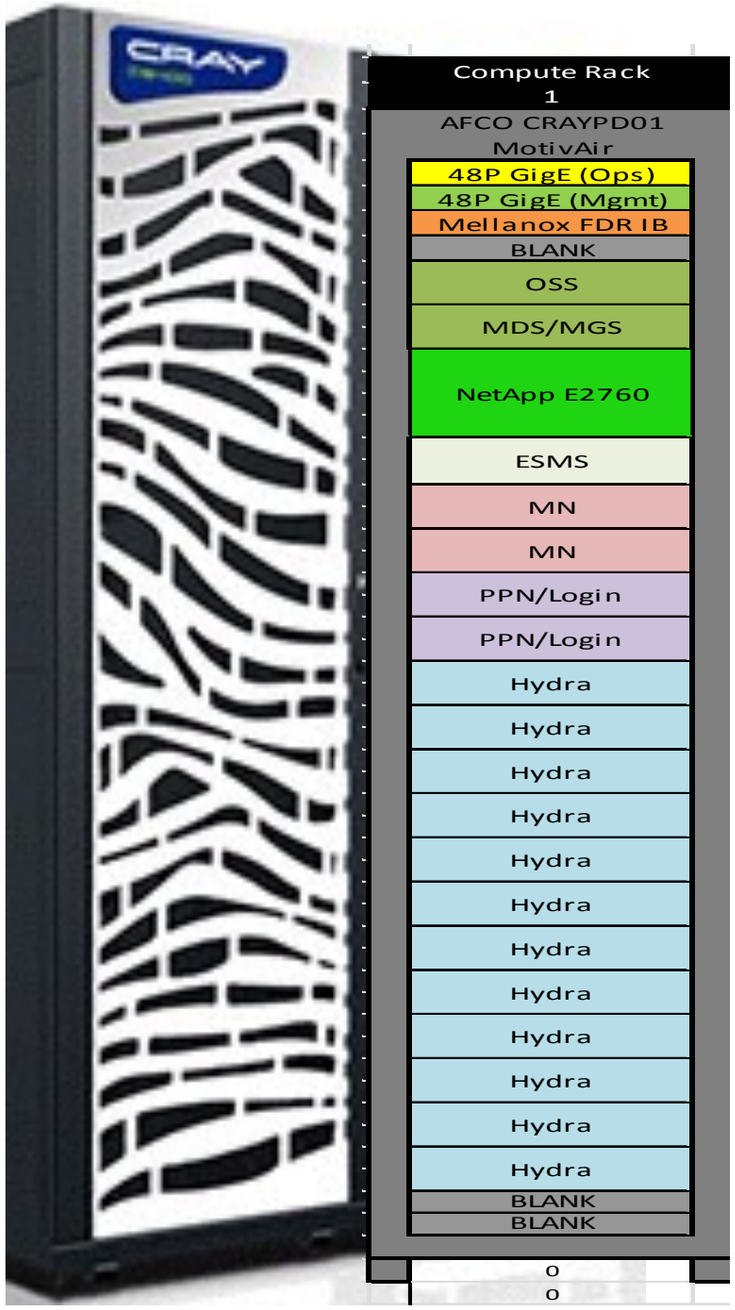
48 CPUs

192 Tesla K80 GPUs

> 90% of FLOPS from GPUs

Operational in 2016

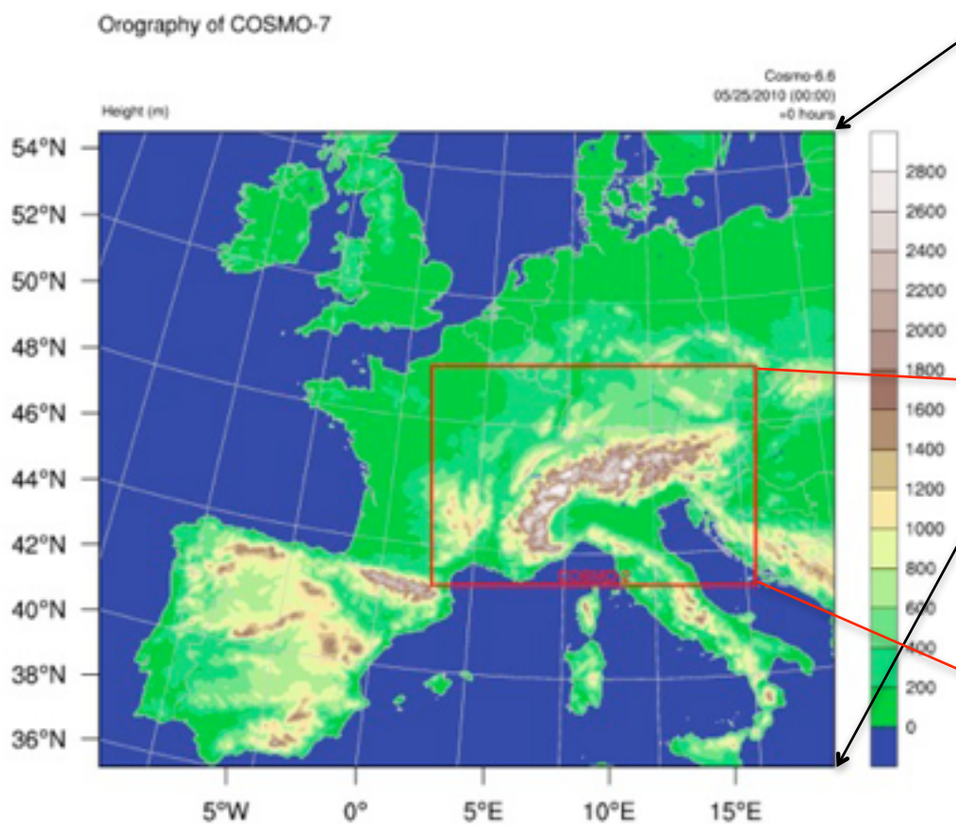
5 NVIDIA

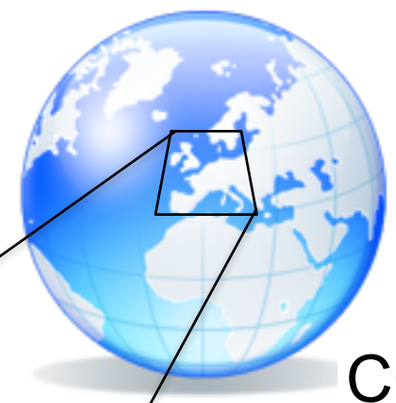


CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Meteo Swiss production suite until March 30, 2016

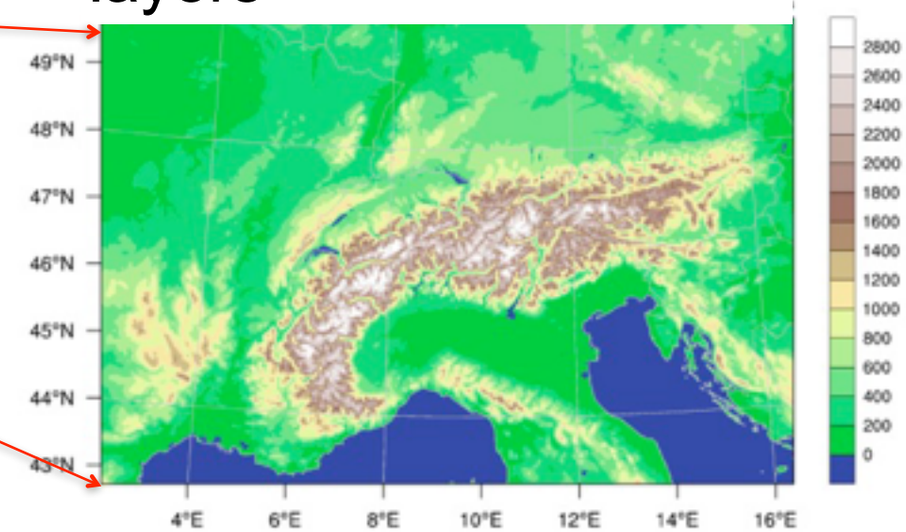
COSMO-7
3x per day 72h forecast
6.6 km lateral grid, 60 layers





ECMWF
2x per day
16 km lateral grid, 91 layers

COSMO-2
8x per day 24h forecast
2.2 km lateral grid, 60 layers



Some of the products generate from these simulations:

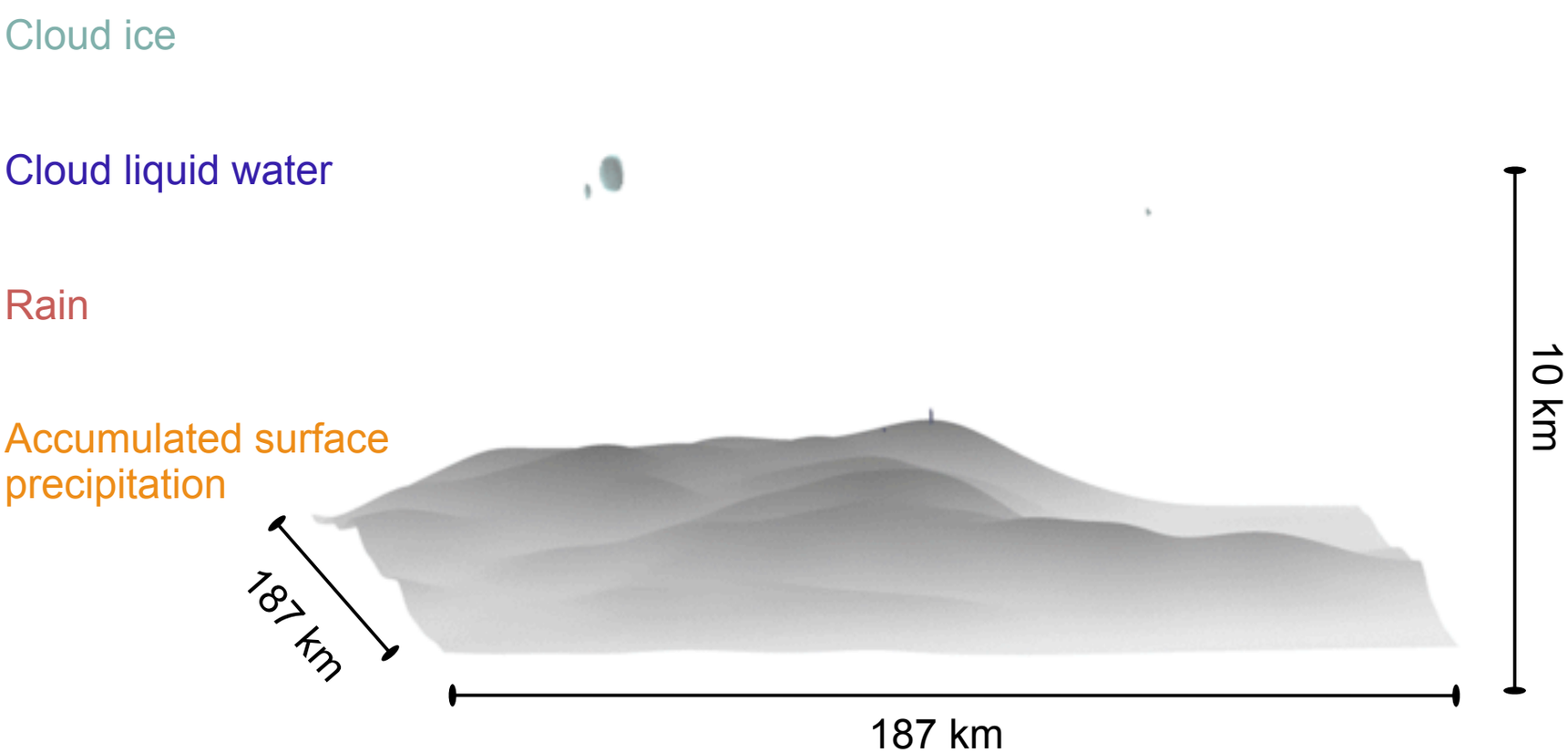
- Daily weather forecast on TV / radio
- Forecasting for air traffic control (Sky Guide)
- Safety management in event of nuclear incidents

“Albis” & “Lema”: CSCS production systems for Meteo Swiss until March 2016

Cray XE6 procured in spring 2012 based on 12-core AMD Opteron multi-core processors

Cloud resolving simulations

Institute for Atmospheric and Climate Science Study at ETH Zürich (Prof. Schär) demonstrates cloud resolving models converge at 1-2km resolution (at least for convective clouds over the alpine region)



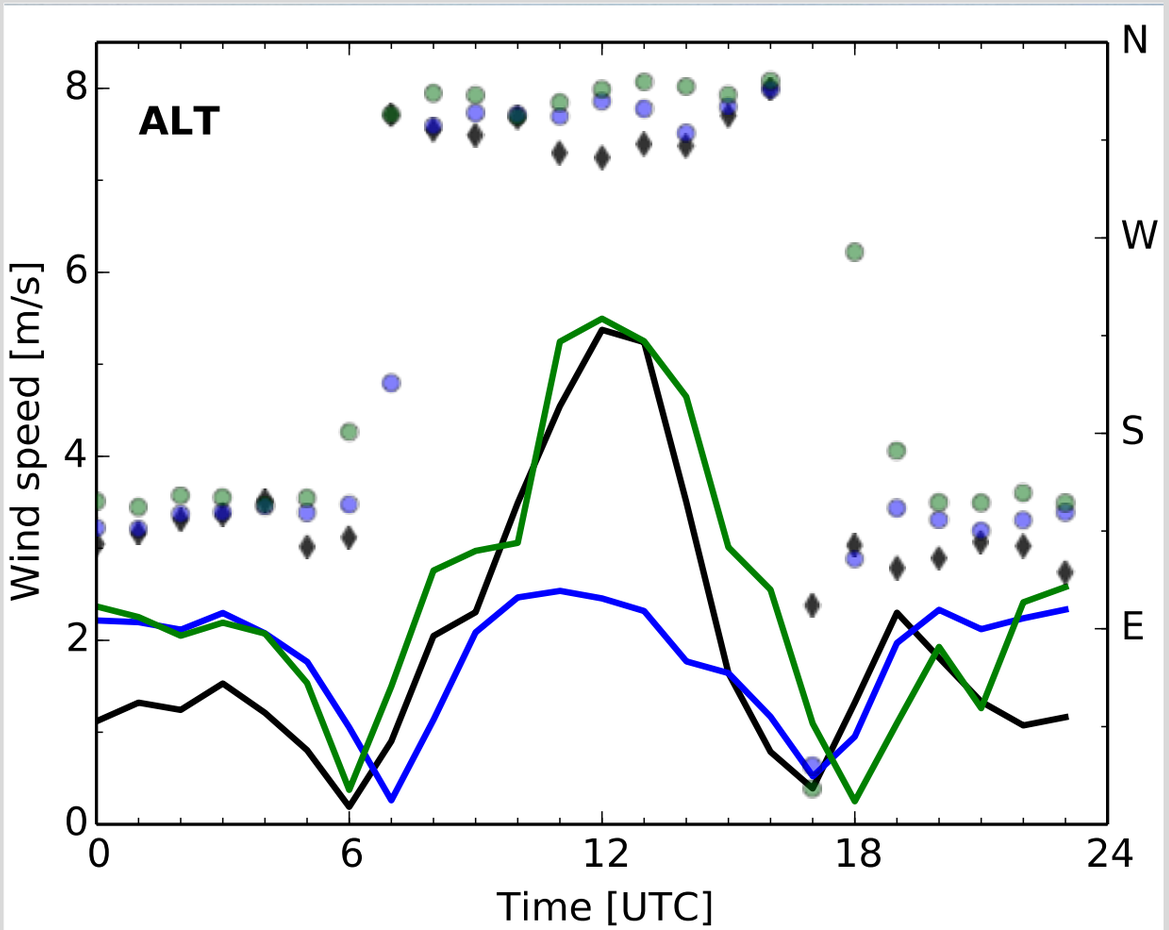
COSMO model setup: Δx=550 m, Δt=4 sec Plots generated using INSIGHT

Orographic convection – simulation: 11-18 local time, 11 July 2006 (Δt_plot=4 min)

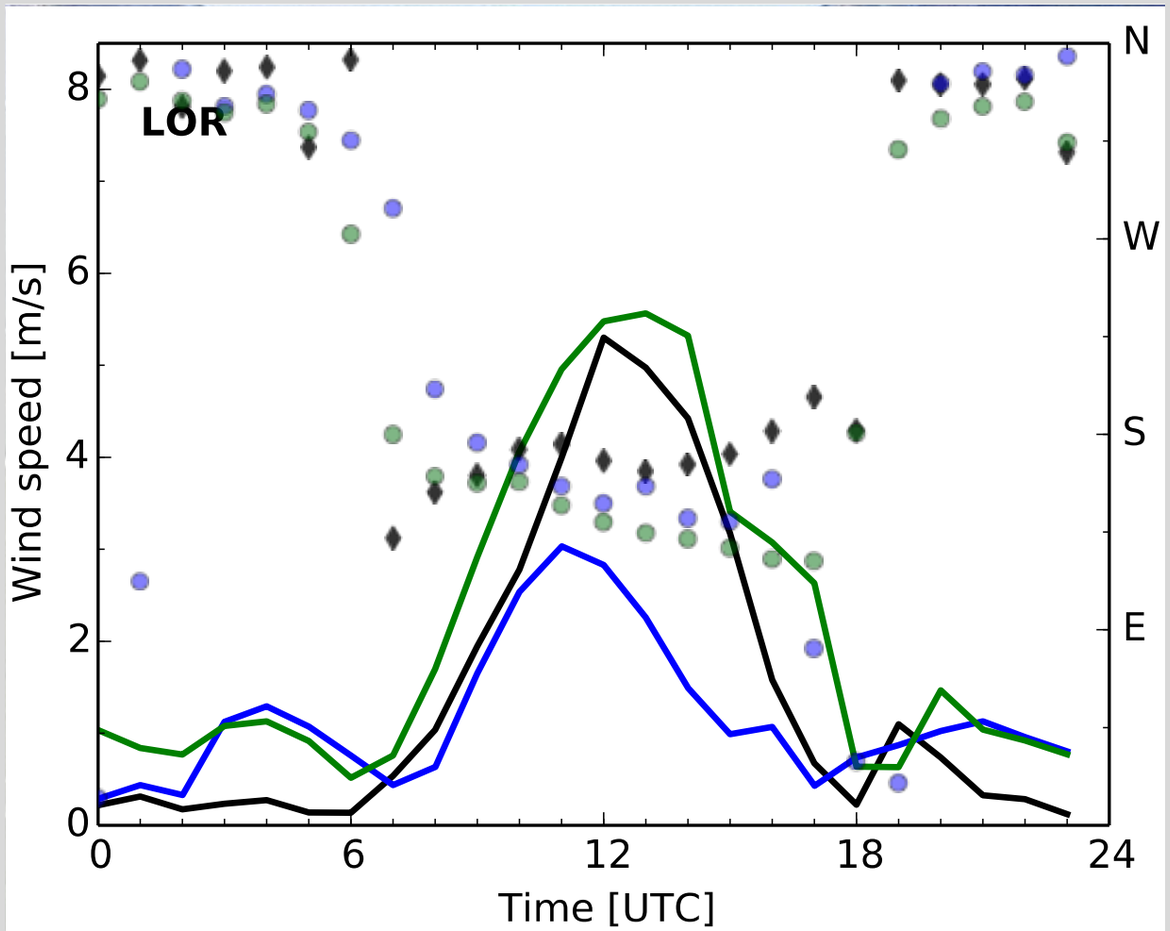
Source: Wolfgang Langhans and Christoph Schär, Institute for Atmospheric and Climate Science, ETH Zurich

Higher resolution is necessary for quantitative agreement with experiment (18 days for July 9-27, 2006)

Altdorf (Reuss valley)



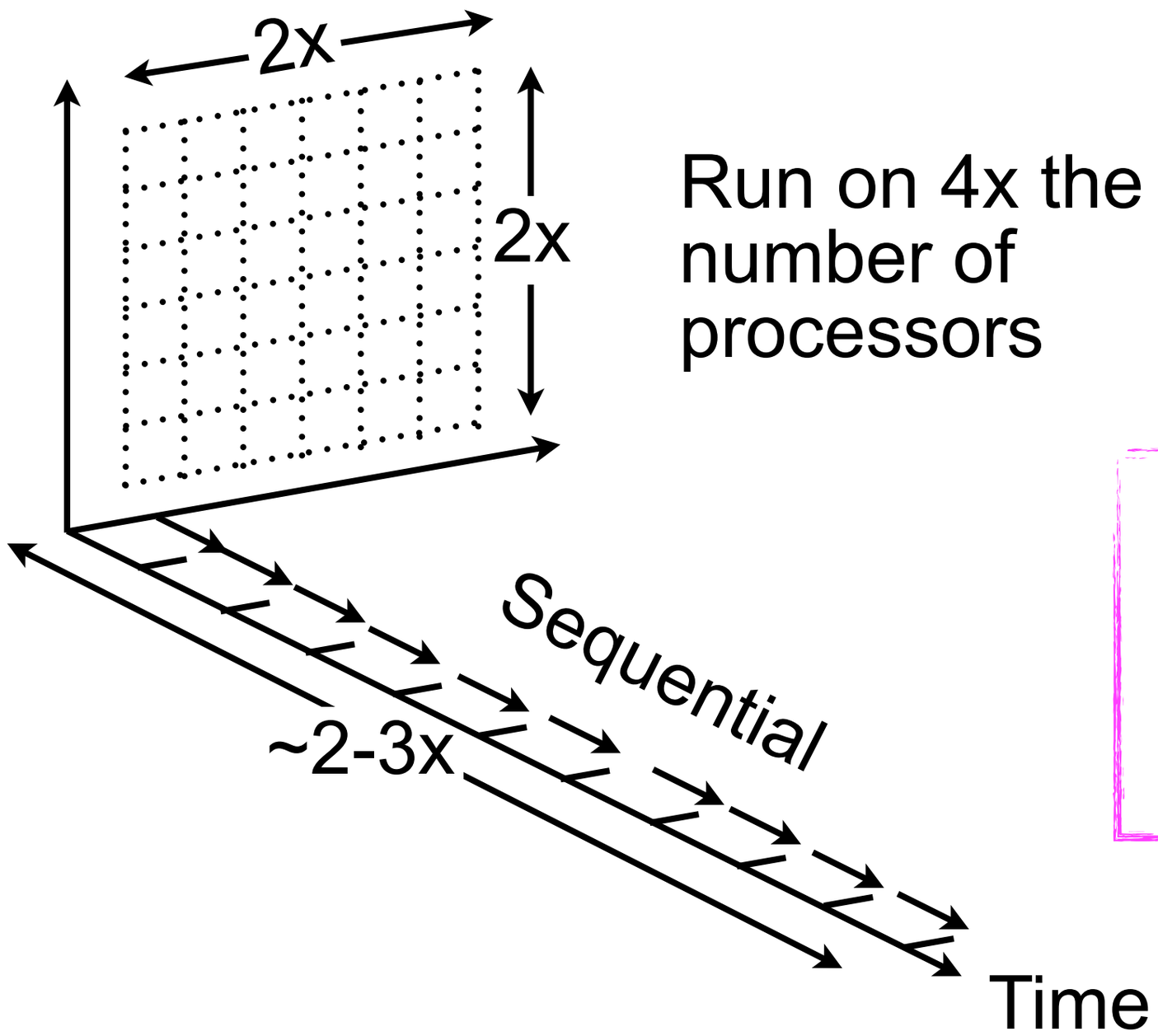
Lodrino (Leventina)



Observation Average wind speed (—) and direction (◊) **COSMO-2** **COSMO-1**

source: Oliver Fuhrer, MeteoSwiss

Improve resolution of Meteo Swiss model from 2 to 1 km

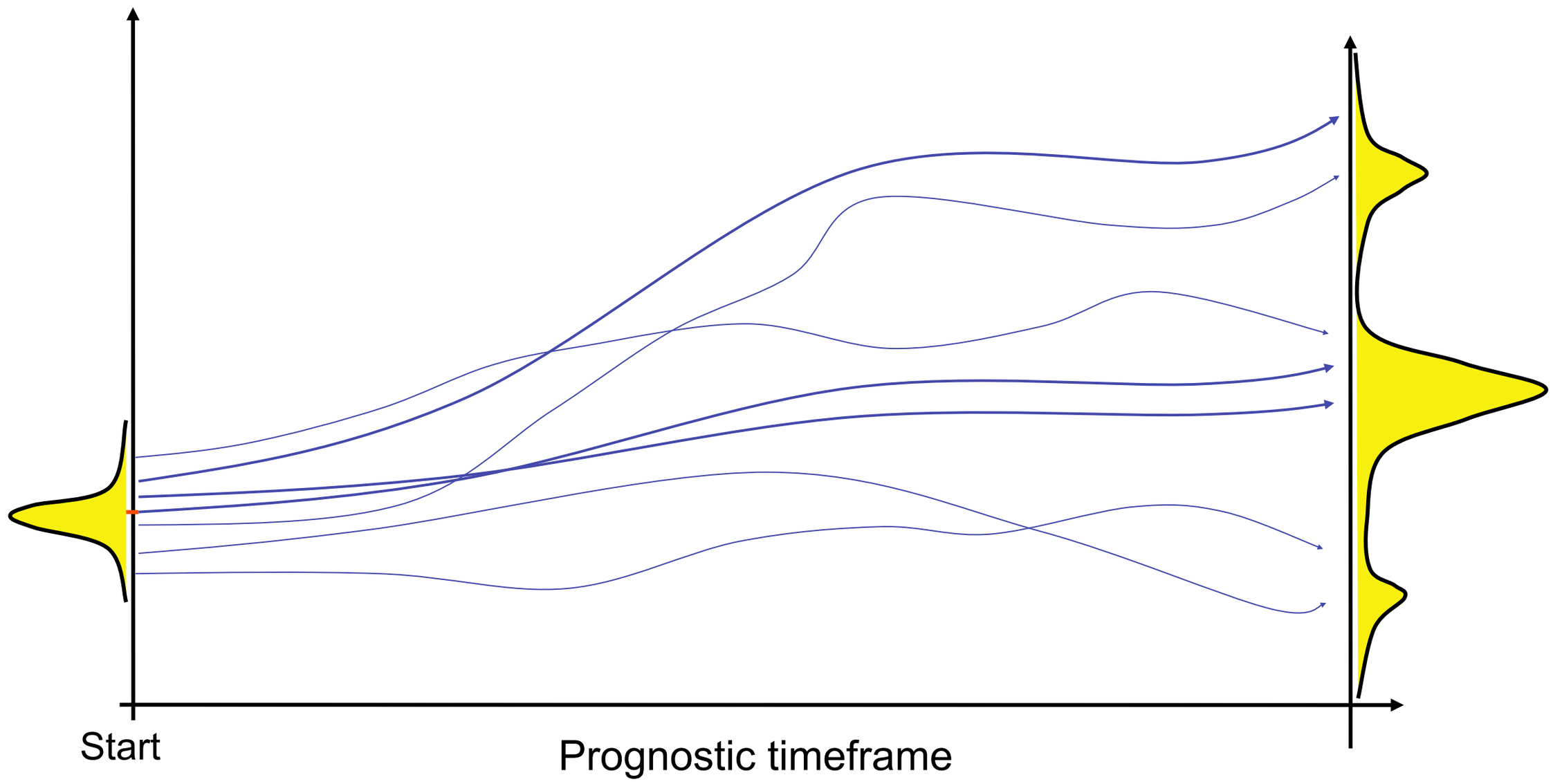


Run on 4x the number of processors

Doubling the resolution requires ~10x performance increase

Prognostic uncertainty

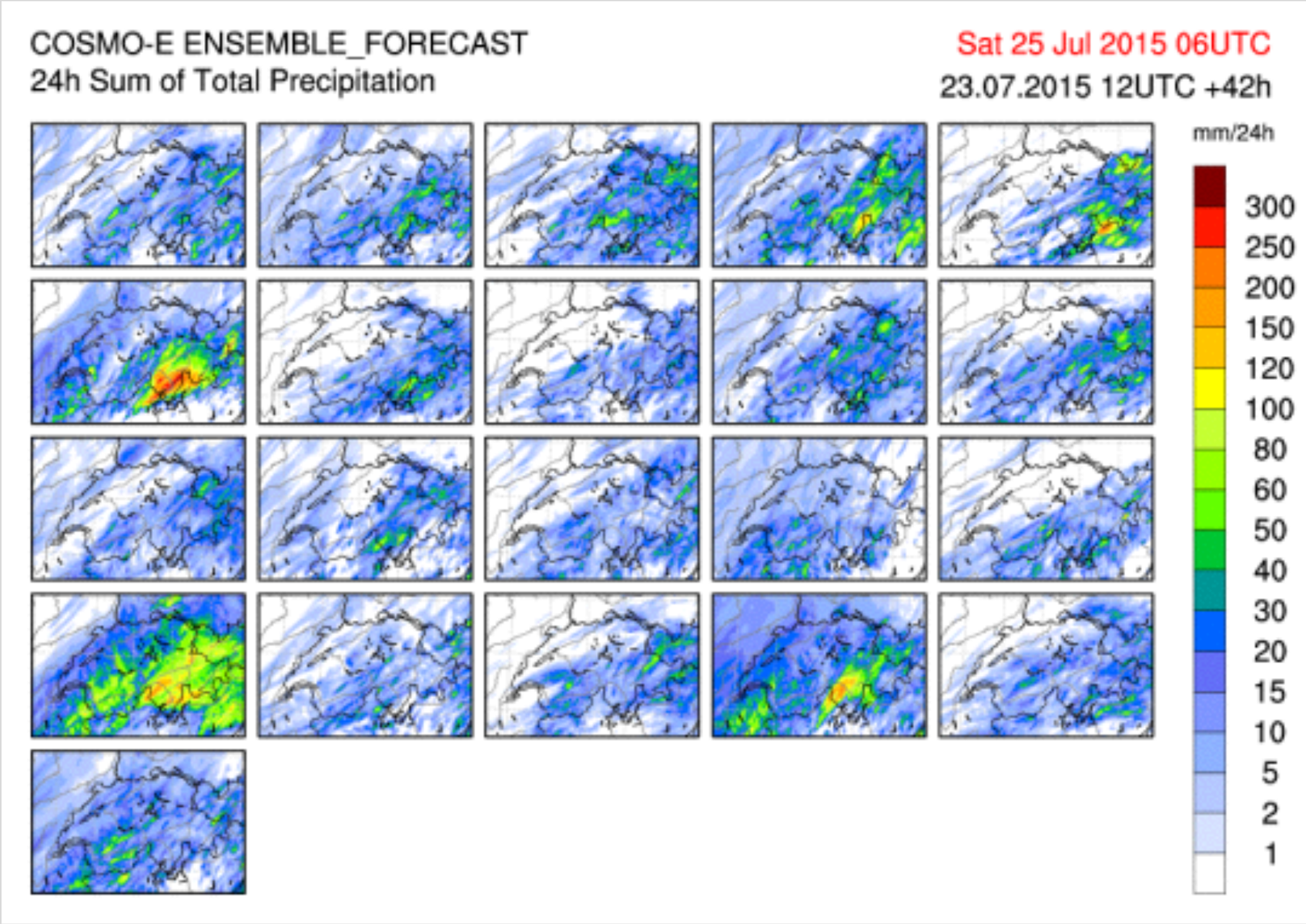
The weather system is chaotic
→ rapid growth of small perturbations (butterfly effect)



Ensemble method: compute distribution over many simulations

Benefit of ensemble forecast

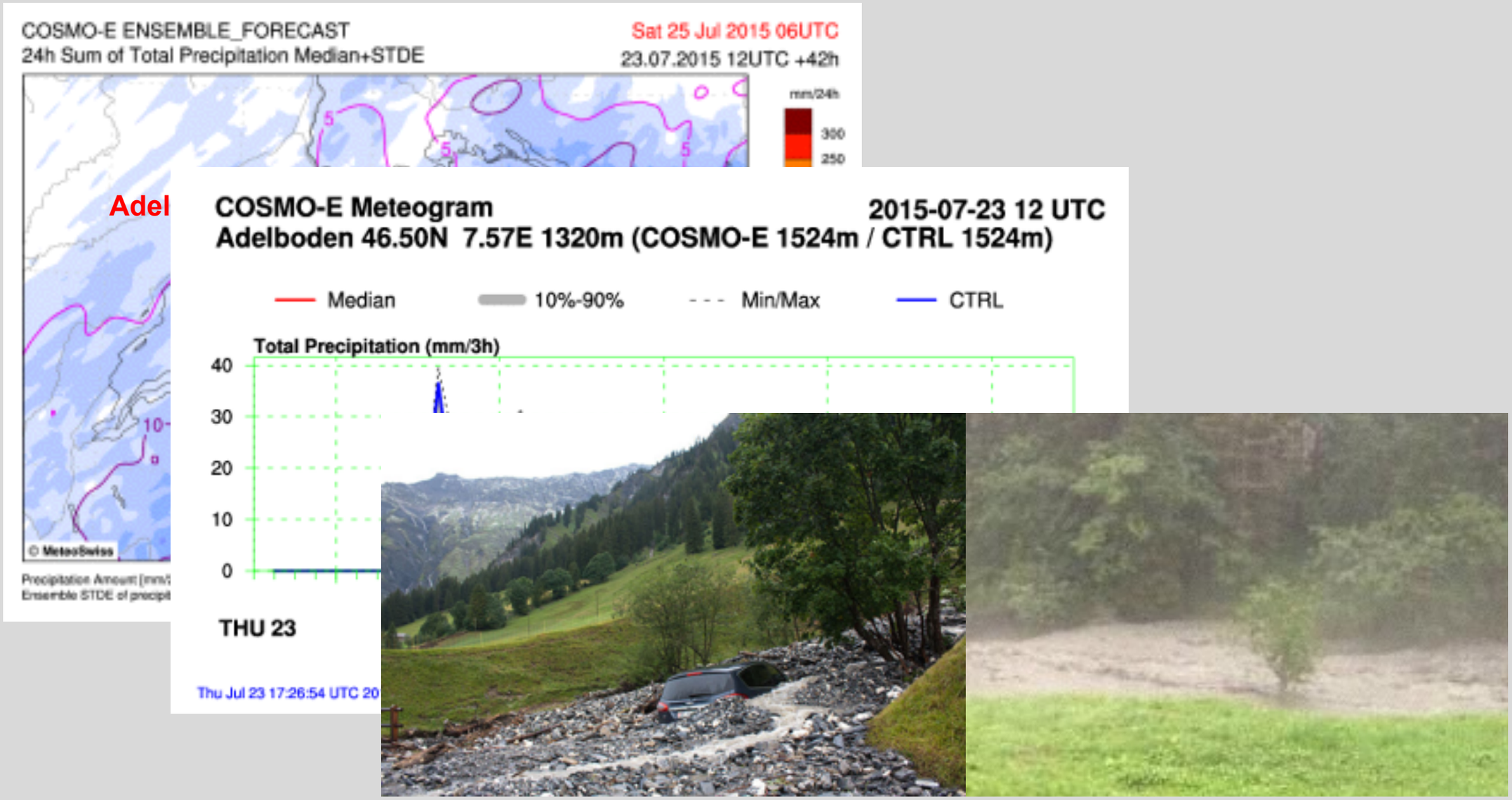
(heavy thunderstorms on July 24, 2015)



source: Oliver Fuhrer, MeteoSwiss

Benefit of ensemble forecast

(heavy thunderstorms on July 24, 2015)



source: Oliver Fuhrer, MeteoSwiss

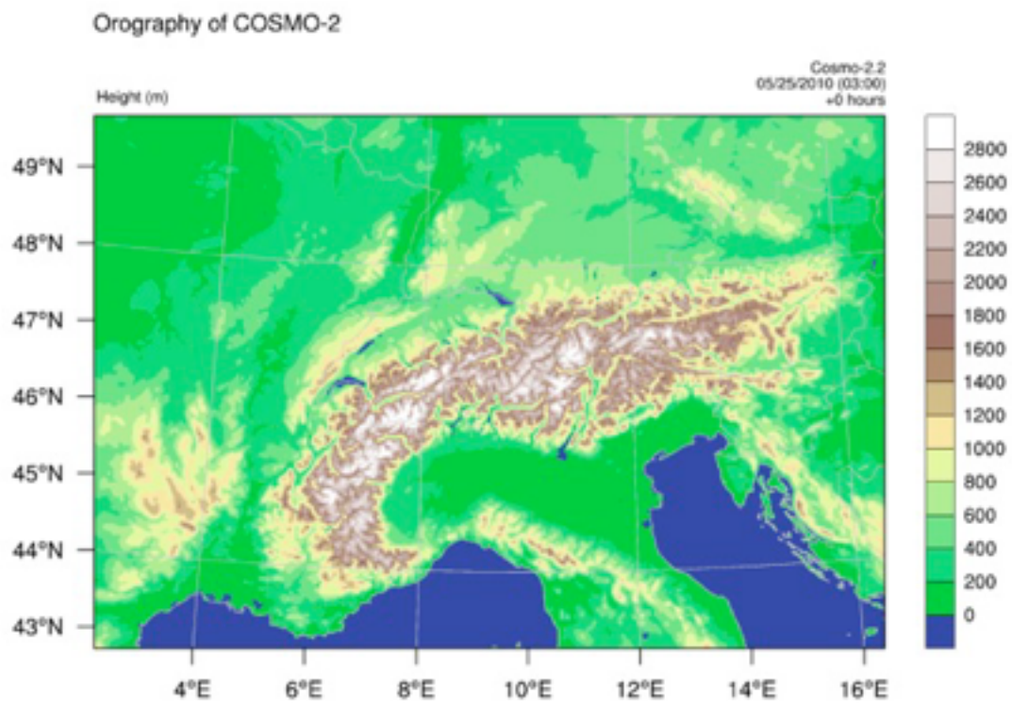
ETH zürich

Improving simulation quality requires higher performance – what exactly and by how much?

Resource determining factors for Meteo Swiss' simulations

Current model running through mid 2016

COSMO-2: 24h forecast running in 30 min.
8x per day



New model starting operation on in Jan. 2016

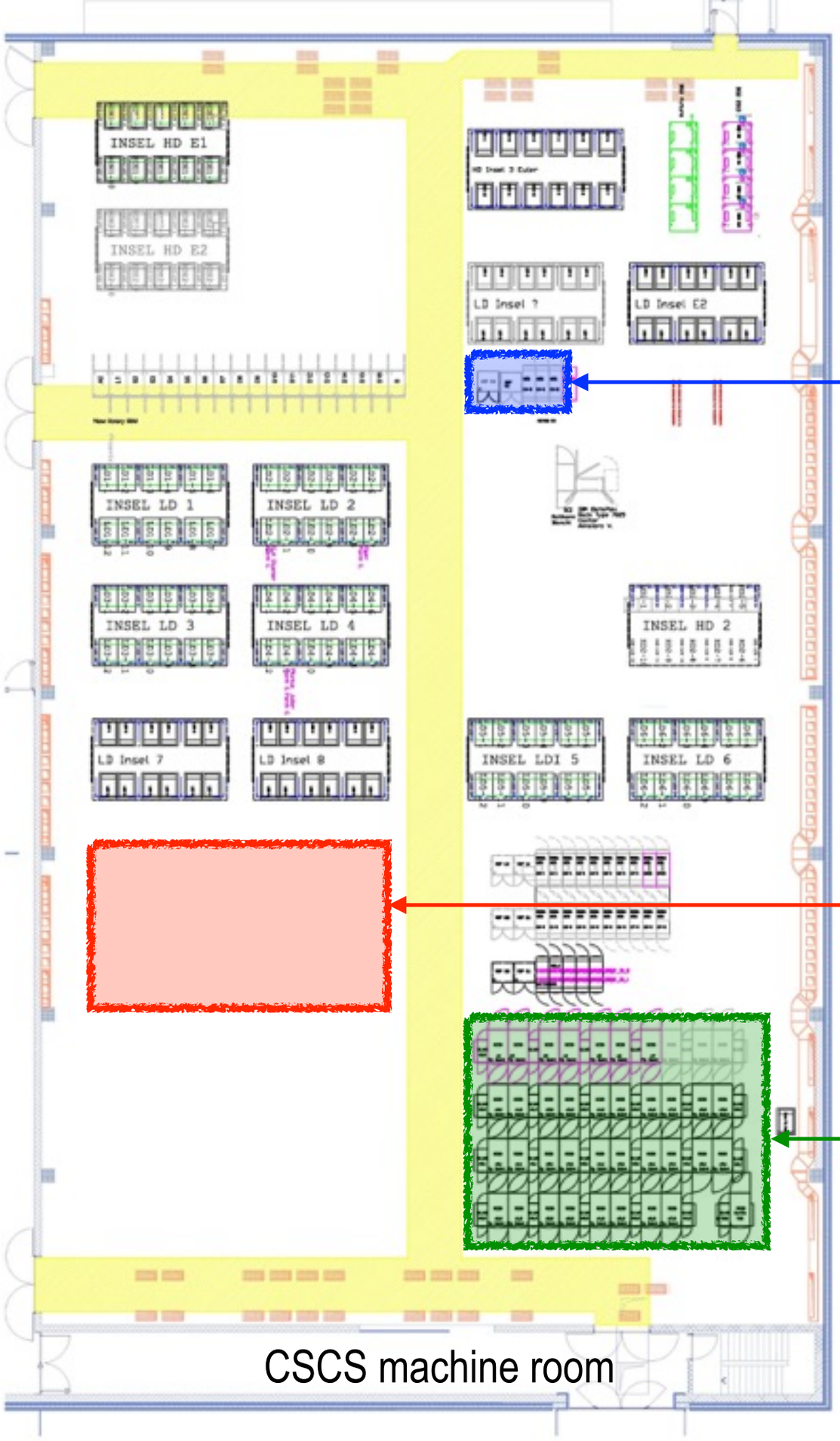
COSMO-1: 24h forecast running in 30 min.
8x per day (**~10x** COSMO-2)

COSMO-2E: 21-member ensemble,120h forecast
in 150 min., 2x per day (**~26x** COSMO-2)

KENDA: 40-member ensemble,1h forecast
in 15 min., 24x per day (**~5x** COSMO-2)

New production system must deliver
~40x the simulations performance
of "Albis" and "Lema"

State of the art implementation of new system for Meteo Swiss



Albis & Lema: 3 cabinets Cray XE6 installed Q2/2012

- New system needs to be installed Q2-3/2015
- Assuming 2x improvement in per-socket performance: ~20x more X86 sockets would require 30 Cray XC cabinets

New system for Meteo Swiss if we build it like the German Weather Service (DWD) did theirs, or UK Met Office, or ECMWF … (30 racks XC)

Current Cray XC30/XC40 platform (space for 40 racks XC)

Thinking inside the box is not a good option!

Co-Design our way out?

Potential for co-design

- Time-to-solution driven (specs are clear)
- Exclusive usage
- Only one performance critical application
- Stable configuration (code & system)
- Current code can be improved
- Novel hardware has yet to be exploited

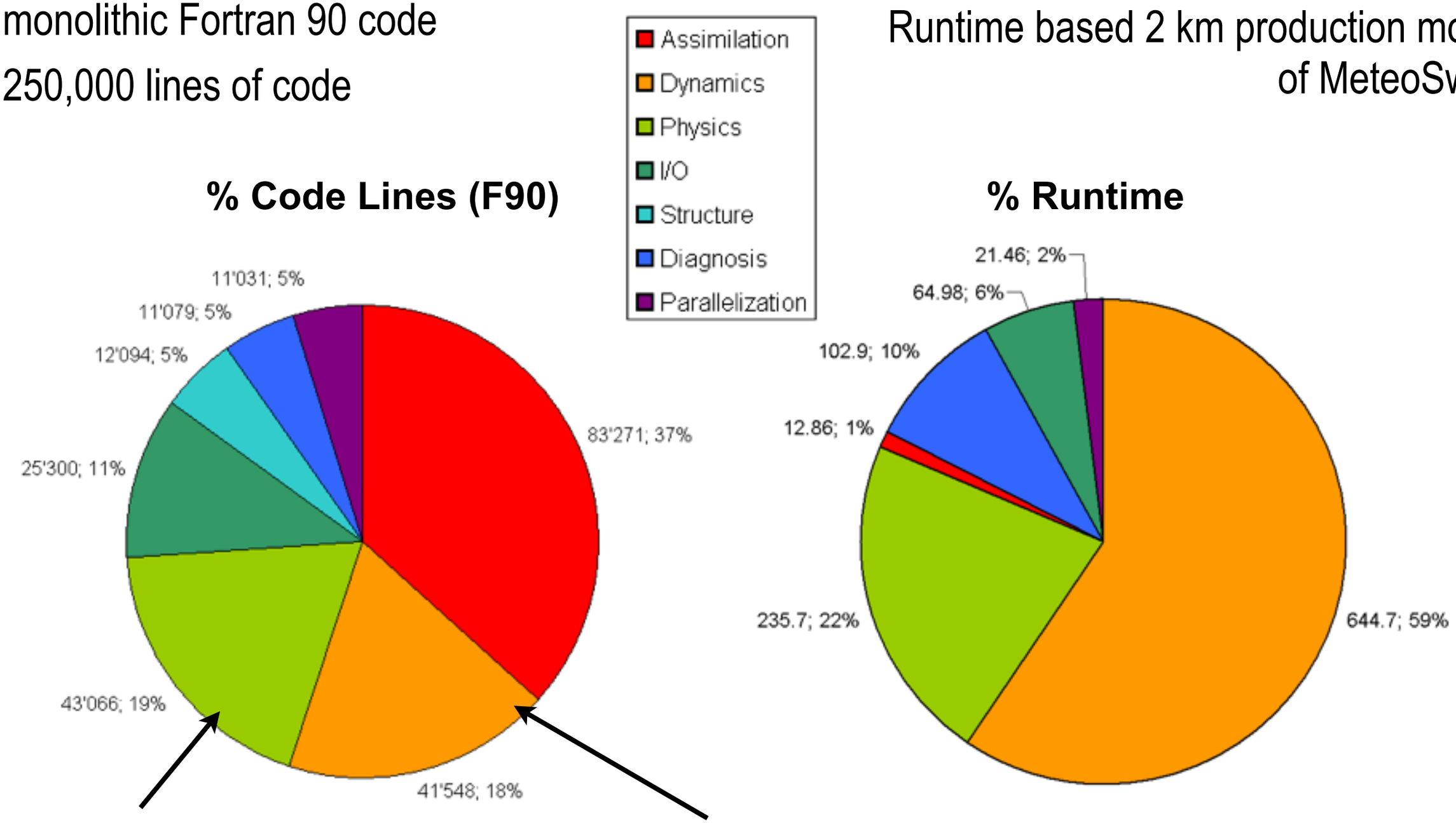
Challenges for making it work

- Community code
  - Large user base
  - Performance portability
  - Knowhow transfer
- Complex workflow
- High reliability required
- Rapidly evolving technology (hardware and software)

COSMO-OPCODE: a legacy code migration project

- monolithic Fortran 90 code
- 250,000 lines of code

Runtime based 2 km production model of MeteoSwiss

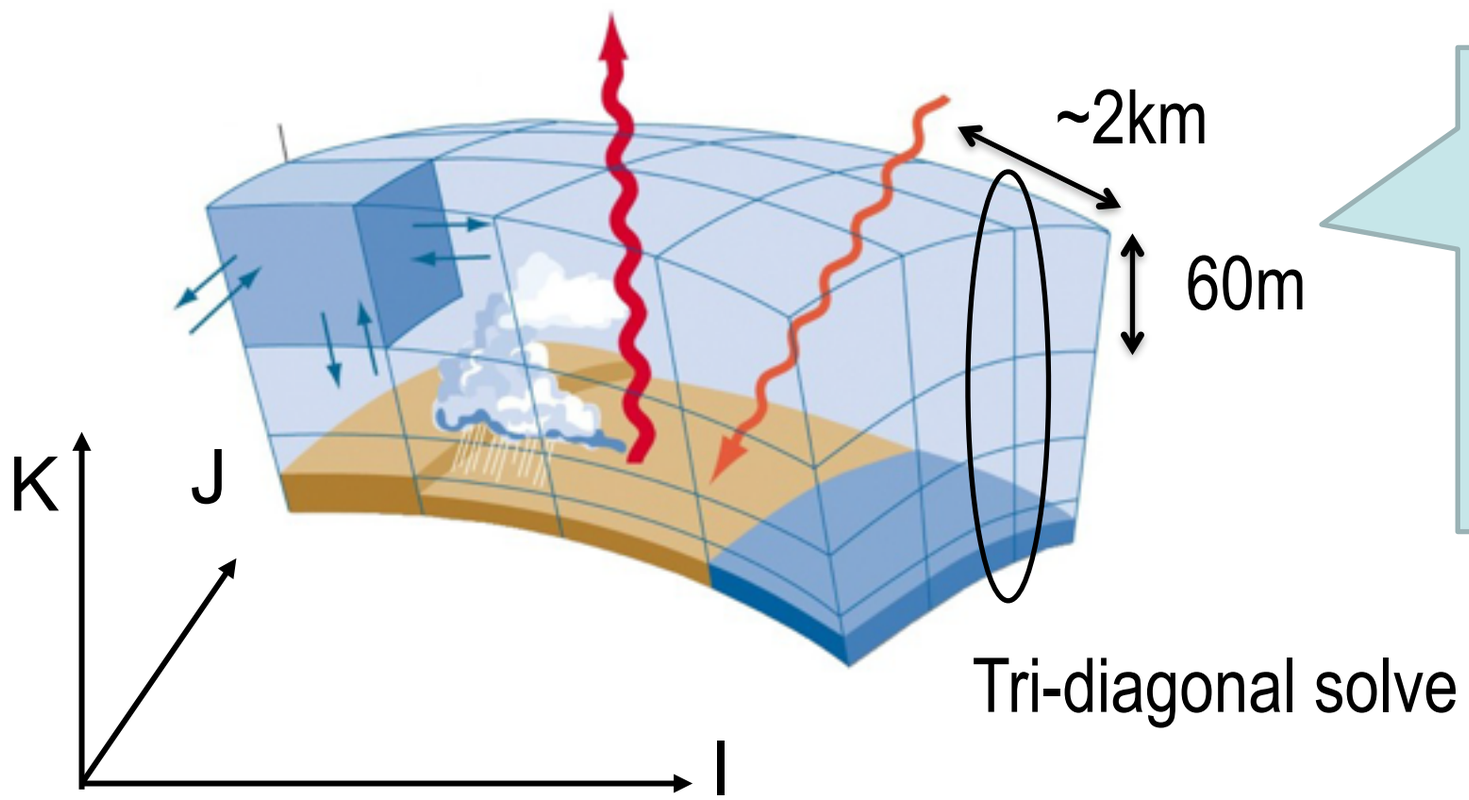


Original code (with OpenACC for GPU)

Rewrite in C++ (with CUDA backend for GPU)

Insight into algorithms and data structures

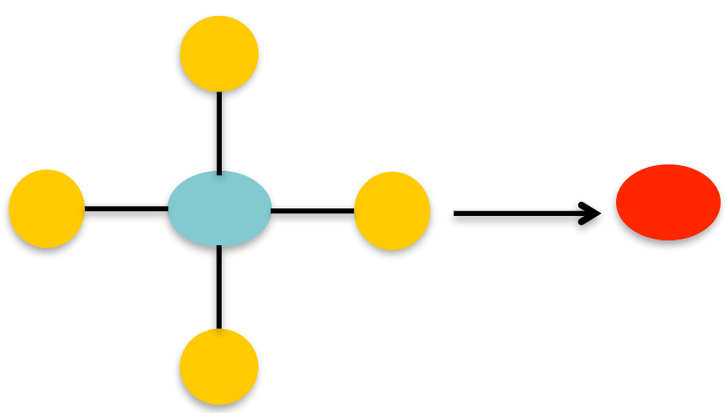
- Partial differential equations solved on structured gird (variables: velocity, temperature, pressure, humidity, etc.)
- Explicit solve horizontally (I,J) using finite difference stencils
- Implicit solve in vertical direction (K) with tri-diagonal solve in every column (applying Thomas algorithm in parallel – can be expressed as stencil)



Due to implicit solves in the vertical we can work with longer time steps (2km and not 60m grid size is relevant)

Stencil example: Laplace operator in 2D

```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```



```
do k = kstart, kend
  do j = jstart, jend
    do i = istart, iend
      lap(i, j, k) = -4.0 * data(i,   j   , k) + &
        data(i+1, j, , k) + data(i-1, j   , k) + &
        data(i  , j+1, k) + data(i  , j-1, k)
    end do
  end do
end do
```

Domain specific library (language)

- A language or library that exploits domain or data structure specific information to achieve optimal performance and productivity
- Domain specific languages come in a number of flavours
  - Source-to-source translation (PATUS, Liszt, ATMO, ICON, …)
  - Compiler extensions (Pochoir, Halide, …)
  - Embedded library into host language (STELLA, Physis, …)
- STELLA (Stencil Loop Language) is an embedded library into C++/STL for solving PDEs with finite differences on a cartesian grid
  - embedded in C++ STL using template meta-programming

Two main components of an operator on a structured grid

1. **Loop-logic** defines stencil application domain and order
2. **Stencil** defines the operator to be applied

```
do k = kstart, kend
  do j = jstart, jend
    do i = istart, iend
      lap(i, j, k) = -4.0 * data(i,   j   , k) + &
        data(i+1, j, , k) + data(i-1, j   , k) + &
        data(i   , j+1, k) + data(i   , j-1, k)
    end do
  end do
end do
```

```
enum { data, lap };

template<typename TEnv>
struct Laplace
{
  STENCIL_STAGE(Tenv)
  STAGE_PARAMETER(FullDomain, data)
  STAGE_PARAMETER(FullDomain, lap)

  static void Do()
  {
    lap::Center() =
      -4.0 * data::Center() +
      data::At(iplus1) +
      data::At(iminus1) +
      data::At(jplus1) +
      data::At(jminus1);
  }
};
```

```
IJKRealField lapfield, datafield;
Stencil stencil;

StencilCompiler::Build(
pack_parameters(
    Param<lap, cInOut>(lapfield),
    Param<data, cIn>(datafield)
),
  concatenate_sweeps(
    define_sweep<KLoopFullDomain>(
      define_stages(
        StencilStage<Laplace, IJRangeComplete>()
      )
    )
  )
);

stencil.Apply();
```

Stencil

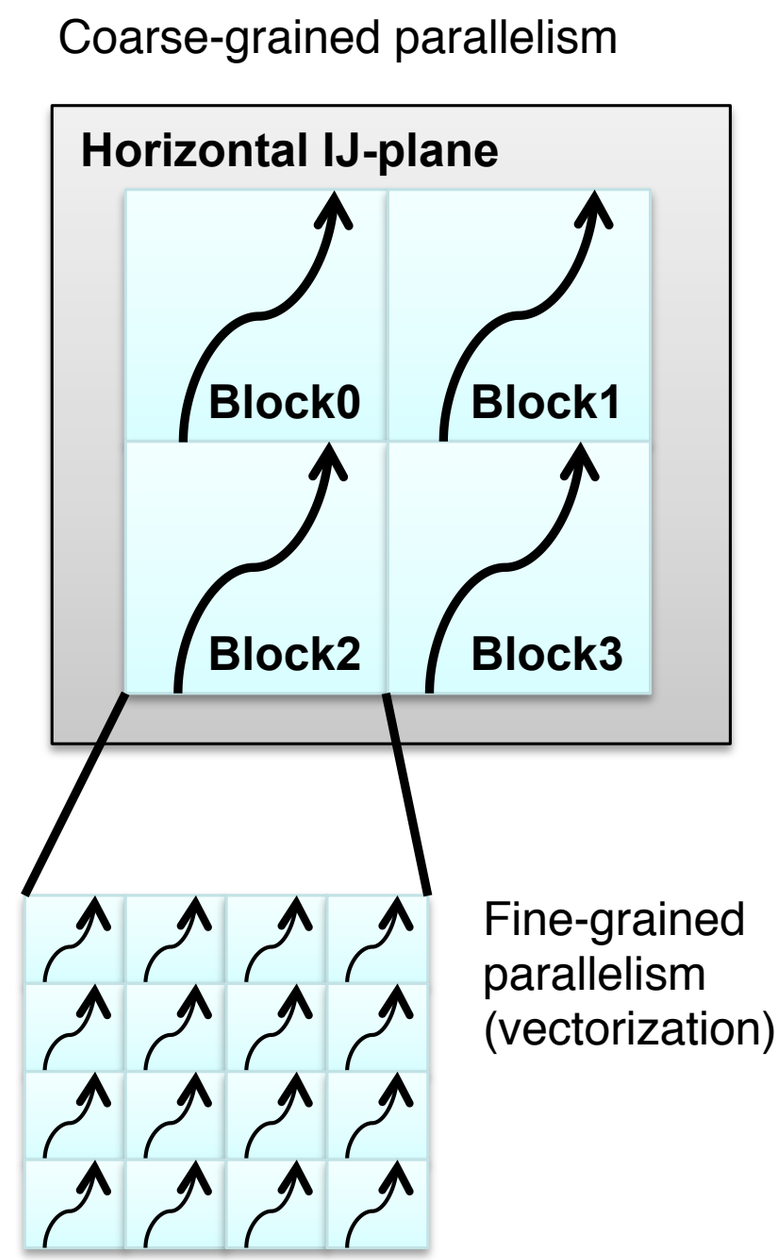
```
enum { data, lap };

template<typename TEnv>
struct Laplace
{
  STENCIL_STAGE(Tenv)
  STAGE_PARAMETER(FullDomain, data)
  STAGE_PARAMETER(FullDomain, lap)

  static void Do()
  {
    lap::Center() =
      -4.0 * data::Center() +
      data::At(iplus1) +
      data::At(iminus1) +
      data::At(jplus1) +
      data::At(jminus1);
  }
};
```

Loop logic

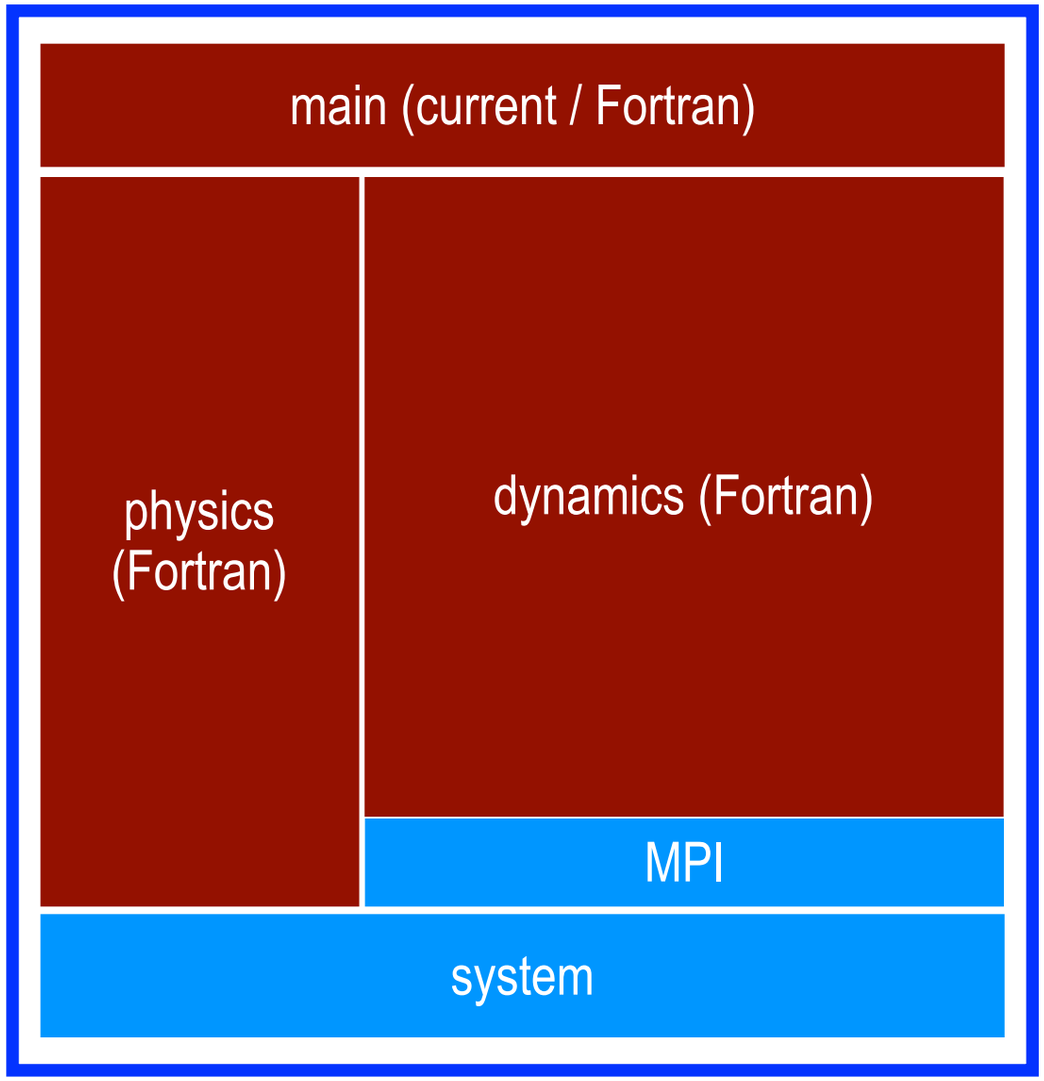
```
IJKRealField lapfield, datafield;
Stencil stencil;

StencilCompiler::Build(
pack_parameters(
    Param<lap, cInOut>(lapfield),
    Param<data, cIn>(datafield)
),
  concatenate_sweeps(
    define_sweep<KLoopFullDomain>(
      define_stages(
        StencilStage<Laplace, IJRangeComplete>()
      )
    )
  )
);

stencil.Apply();
```

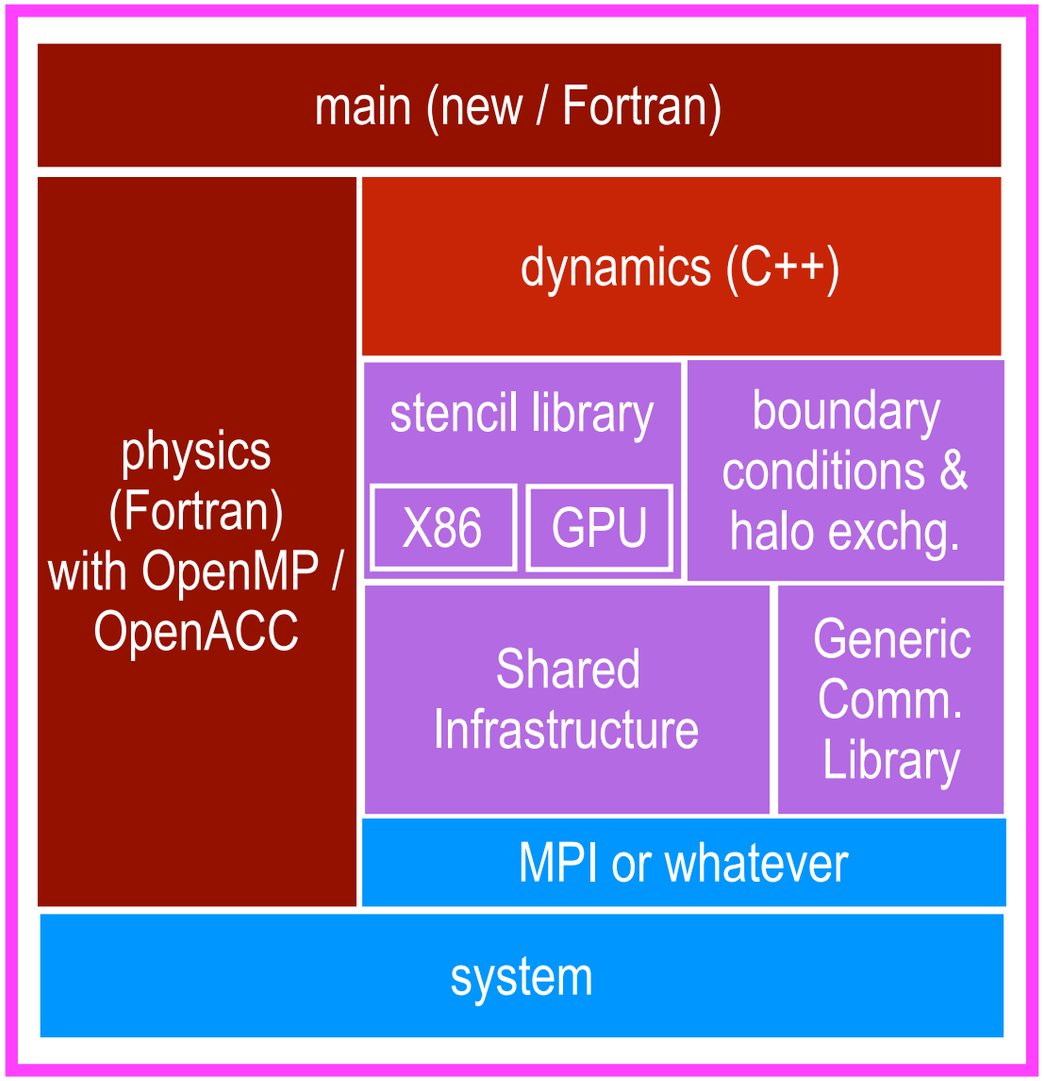
Architecture dependent backend

- The same user-level code can be compiled with different, architecture dependent backends
- **multi-core CPU (x86)**
  - kij-storage
  - ij-blocking
  - Coarse: OpenMP theads
  - Fine: vectorisation by compiler
- **GPU (Tesla)**
  - ijk-storage
  - Coarse: CUDA thread blocks
  - Fine: CUDA threads
  - software managed caching



COSMO: old and new (refactored) code





References and Collaborators

- Peter Messmer and his team at the NVIDIA co-design lab at ETH Zurich
- Teams at CSCS and Meteo Suisse, group of Christoph Schaer @ ETH Zurich
- O. Fuhrer, C. Osuna, X. Lapillonne, T. Gysi, B. Cumming, M. Bianco, A. Arteaga, T. C. Schulthess, "**Towards a performance portable, architecture agnostic implementation strategy for weather and climate models**", Supercomputing Frontiers and Innovations, vol. 1, no. 1 (2014), see superfri.org
- G. Fourestey, B. Cumming, L. Gilly, and T. C. Schulthess, "**First experience with validating and using the Cray power management database tool**", Proceedings of the Cray Users Group 2014 (CUG14) (see arxiv.org for reprint)
- B. Cumming, G. Fourestey, T. Gysi, O. Fuhrer, M. Fatica, and T. C. Schulthess, "**Application centric energy-efficiency study of distributed multi-core and hybrid CPU-GPU systems**", Proceedings of the International Conference on High-Performance Computing, Networking, Storage and Analysis, SC'14, New York, NY, USA (2014). ACM
- T. Gysi, C. Osuna, O. Fuhrer, M. Bianco and T. C. Schulthess, "**STELLA: A domain-specific tool for structure grid methods in weather and climate models**", to be published in Proceedings of the International Conference on High-Performance Computing, Networking, Storage and Analysis, SC'15, New York, NY, USA (2015). ACM

Piz Kesch / Piz Escha: appliance for meteorology

- Water cooled rack (48U)
- 12 compute nodes with
  - 2 Intel Xeon E5-2690v3 12 cores @ 2.6 GHz256 GB 2133 MHz DDR4 memory
  - 8 NVIDIA Tesla K80 GPU
- 3 login nodes
- 5 post-processing nodes
- Mellanox FDR InfiniBand
- Cray CLFS Luster Storage
- Cray Programming Environment

Origin of factor 40 performance improvement

Performance of COSMO running on new "Piz Kesch" compared to (in Sept. 2015)

(1) previous production system – Cray XE6 with AMD Barcelona

(2) "Piz Dora" – Cray XE40 with Intel Haswell (E5-2690v3)

- Current production system installed in 2012
- New Piz Kesch/Escha installed in 2015
  - Processor performance **2.8x** ← Moore's Law
  - Improved system utilisation **2.8x**
  - General software performance **1.7x**
  - Port to GPU architecture **2.3x**

  ← Software refactoring
  - Increase in number of processors **1.3x**
  - Total performance improvement **~40x**
- Bonus: simulation running on GPU is **3x** more energy efficient compared to conventional state of the art CPU

A factor 40 improvement with similar physical footprint and ~30% reduction in energy consumption

Albis & Lema (in production through 3/2016)

New system: Kesch & Escha

Join us @ the PASC16 Conference

PASC16 provides an opportunity for scientists and practitioners to discuss key issues in the use of High Performance Computing (HPC) in branches of science that require computer modelling and simulations.The scientific program will offer invited lectures, minisymposia, contributed talks and poster presentations. The active participation of graduate students and postdocs is strongly encouraged.

PASC16 Platform for Advanced Scientific Computing Conference

Lausanne Switzerland | **08-10 June 2016**

Conference information, registration and submission
www.pasc16.org

Queries may be addressed to
pasc16@pasc-ch.org

Venue
EPF Lausanne
Swiss Tech Convention Center
Lausanne
Switzerland

Contributions

Researchers from the academic and from the corporate world are invited to participate and present their research area in the form of minisymposia, contributed talks and/or poster presentations. PASC16 welcomes submissions in the following scientific fields:

- CLIMATE & WEATHER
- SOLID EARTH
- LIFE SCIENCE
- CHEMISTRY & MATERIALS
- PHYSICS
- COMPUTER SCIENCE & MATHEMATICS
- ENGINEERING
- EMERGING DOMAINS

Abstracts should describe original, interesting, and solid scientific content that is relevant to computational sciences and HPC. Cross-disciplinary approaches are highly encouraged.

Organization Committee

Maria Grazia Giuffreda, ETH Zurich / CSCS
Jan Hesthaven, EPFL
Torsten Hoefler, ETH Zurich
David Keyes, KAUST
Nicola Marzari, EPFL
Olaf Schenk, USI
Thomas Schulthess, ETH Zurich / CSCS