ETH zürich

Exascale Computing: What are the Goal and Baseline?

Thomas C. Schulthess

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

One of the drivers in U.S. scientific computing: the National Strategic Computing Initiative

Executive Order (7/29/2015)

Goal: Sustain/enhance U.S. leadership in HPC technology and use

How:
1. Use HPC for economic competitiveness and scientific discovery
2. Foster public-private collaboration (all industry, not just vendors)
3. Use a whole-of-government approach (inter-agency collaboration)
4. Move HPC research into production settings

Strategic objectives:
1. Accelerating delivery of a capable exascale computing system across a range of apps. rep. gov. needs
2. Increasing coherence between modelling and simulation and that used for data analytic computing
3. Establishing, over the next 15 years, a viable path forward for HPC systems even beyond limits of CMOS
4. Increasing the capacity and capability of an enduring national HPC ecosystem
5. Developing an enduring public-private collaboration

Lead agencies:

DOE – exascale computing program to support simulations & analytics

NSF – HPC ecosystem for science; workforce development

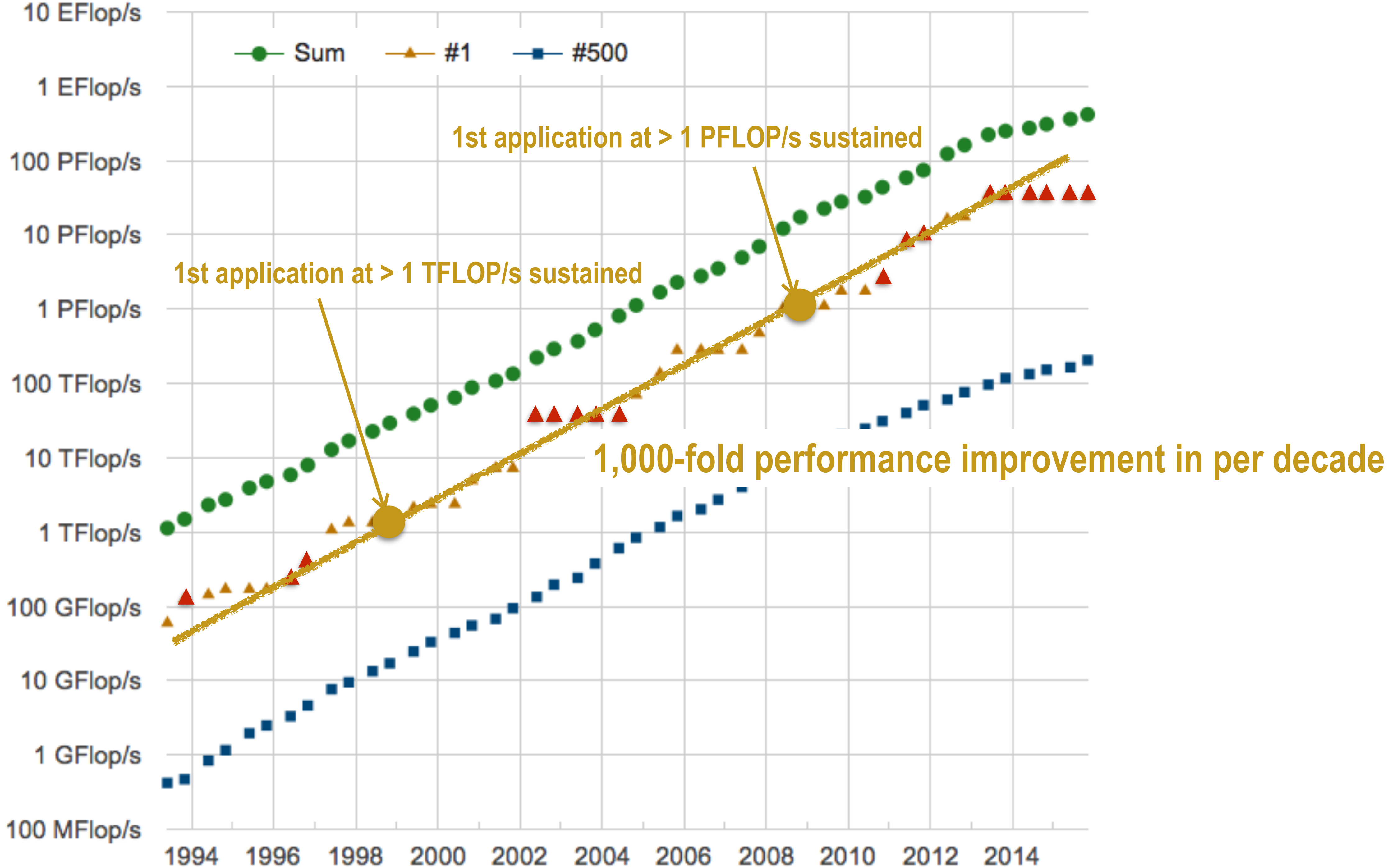
DOD – focus on advanced analytics in support of its mission

this summary by: Steve Conway, IDC

What exactly is our metric for exascale computing?

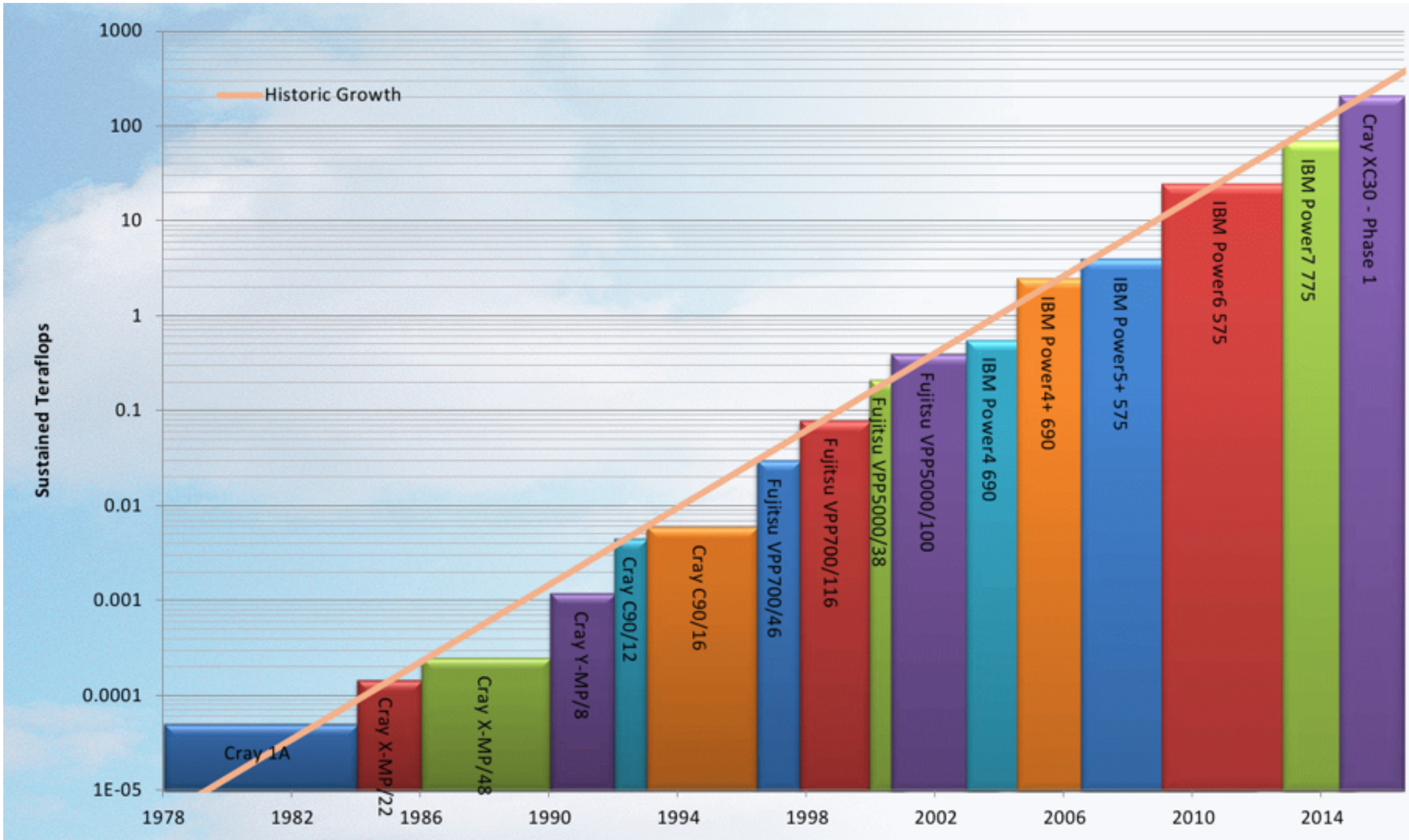
- Today, the fastest supercomputers sustain 20-100 petaflops on HPL
- Thus, a sustained exaflops would be a factor 10-50 away from today's fastest supercomputer
  - There is a questions about productivity of these fastest systems
- Thus, let's be conservative an agree on exascale computing being a **factor 100 more** in sustained application performance over today's (2016) best capabilities

Linpack benchmark solves: $Ax = b$



for the historic development of supercomputing performance, see www.top500.org

"only" 100-fold performance improvement for climate codes

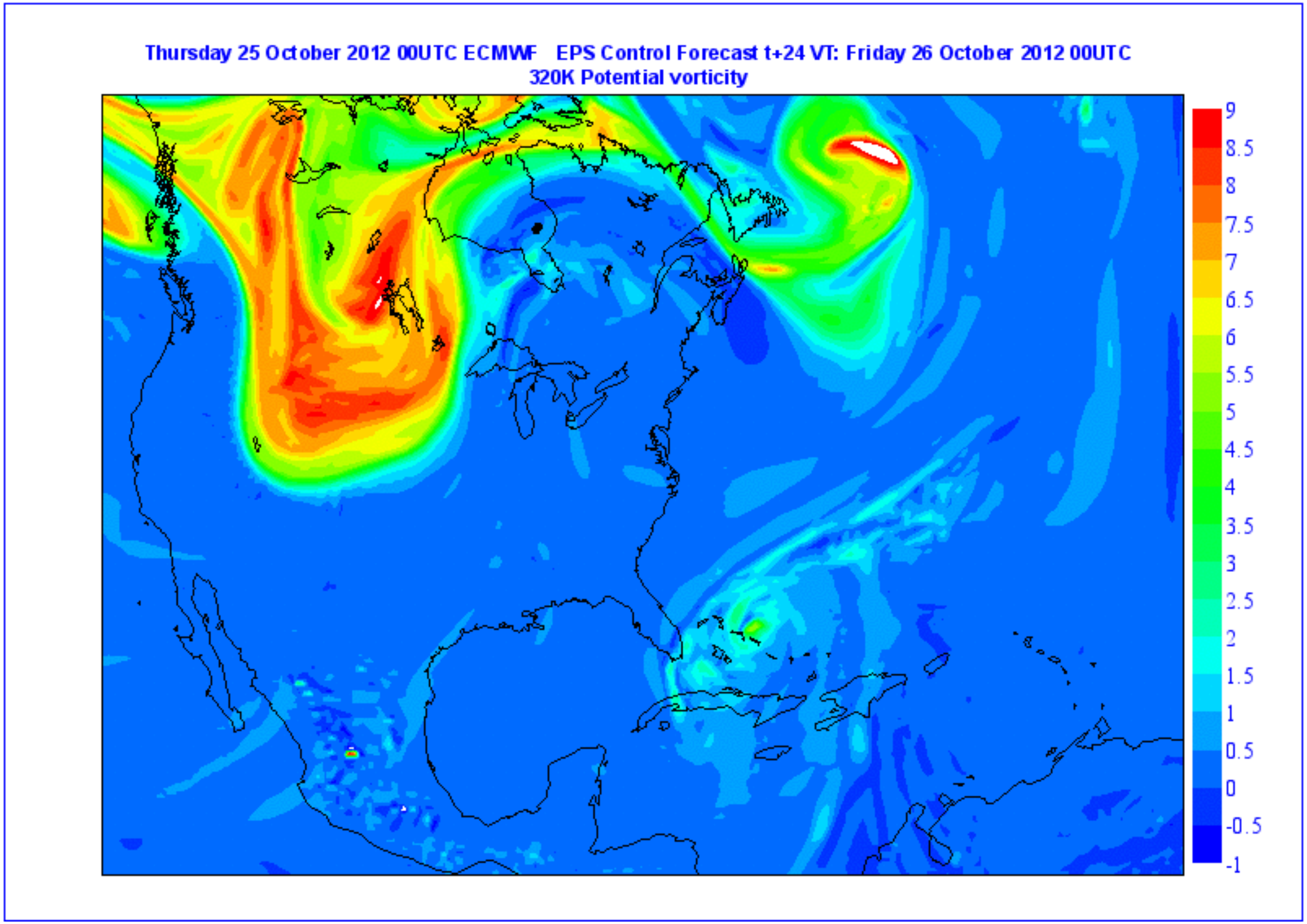


Source: Peter Bauer, ECMWF

Has efficiency of climate codes dropped 10-fold every decade decade?

ETH zürich

Application performance must be factored into the metric for supercomputing at exascale



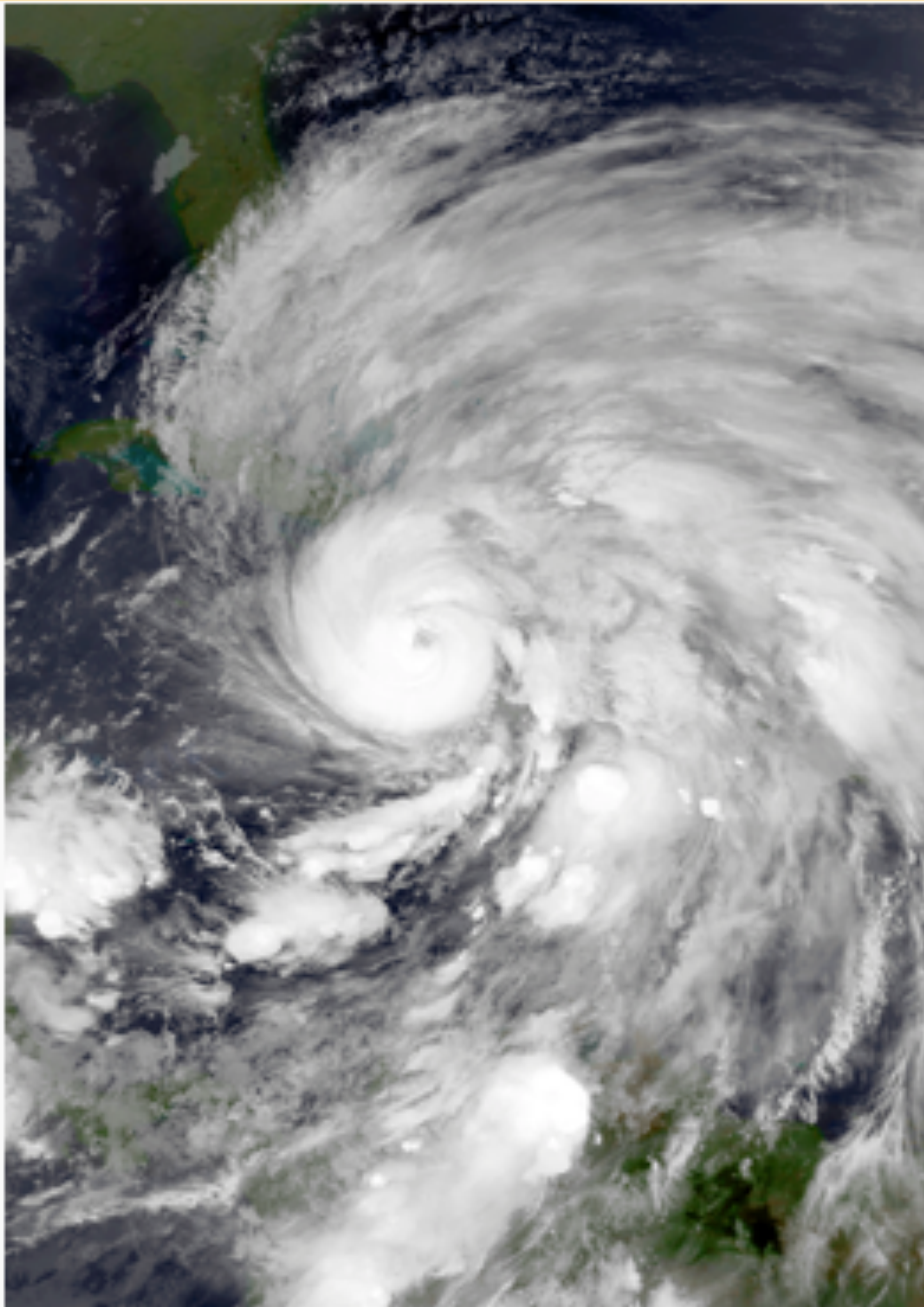
source: http://www.ecmwf.int/en/about/media-centre/news/2013/ecmwf-forecast-data-hurricane-sandy-available-researchers

ETH zürich

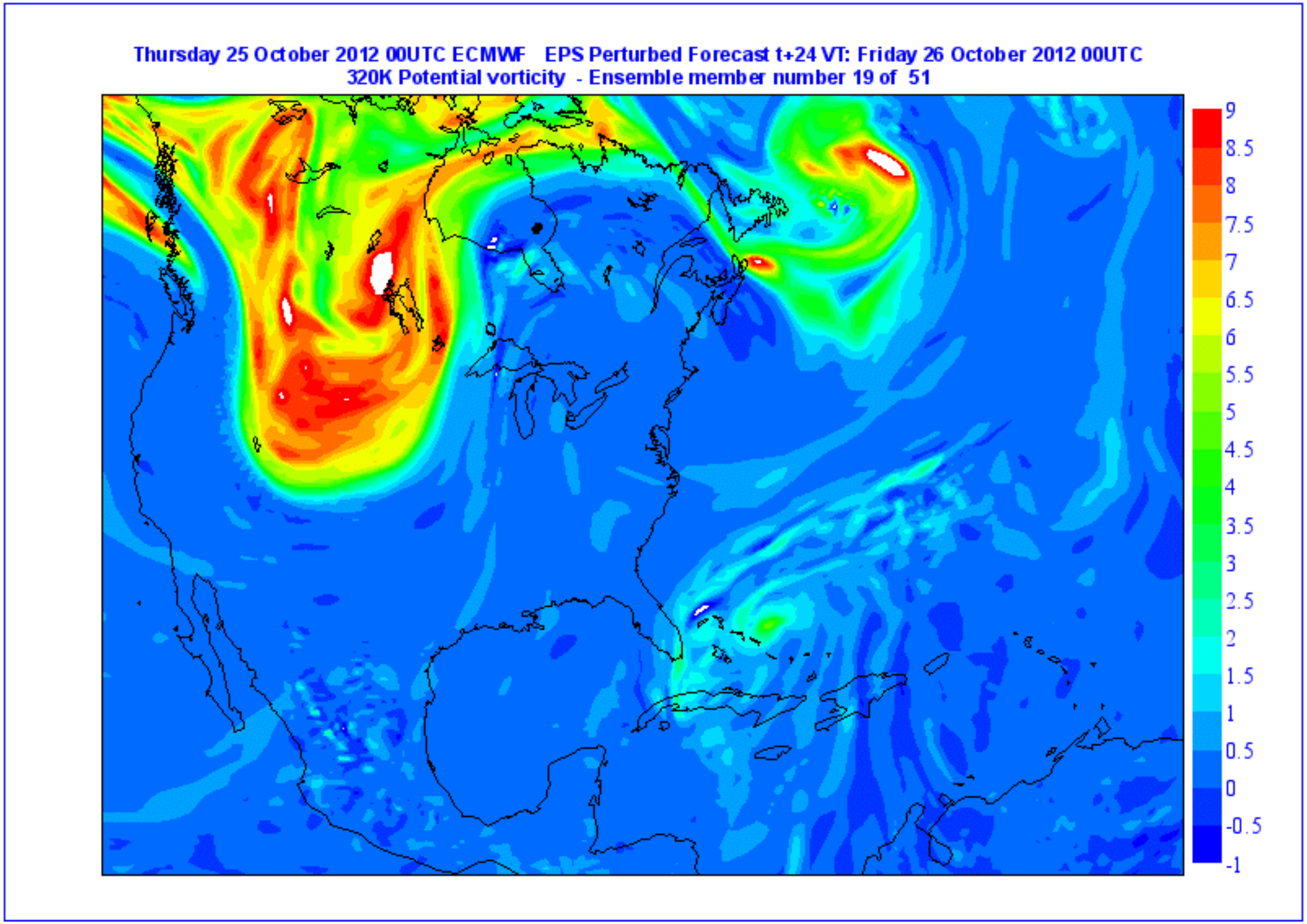
Hurricane Sandy as a Category 3 hurricane on October 25, 2012

Formed	October 22, 2012
Dissipated	November 2, 2012[1] (Extratropical after October 29)
Highest winds	*1-minute sustained:* 115 mph (185 km/h)
Lowest pressure	940 mbar (hPa); 27.76 inHg
Fatalities	233 total (direct and indirect)[2]
Damage	$75 billion (2012 USD) (Second-costliest hurricane in U.S. history[1])
Areas affected	Greater Antilles, Bahamas, most of the eastern United States (especially the coastal Mid-Atlantic States), Bermuda, eastern Canada

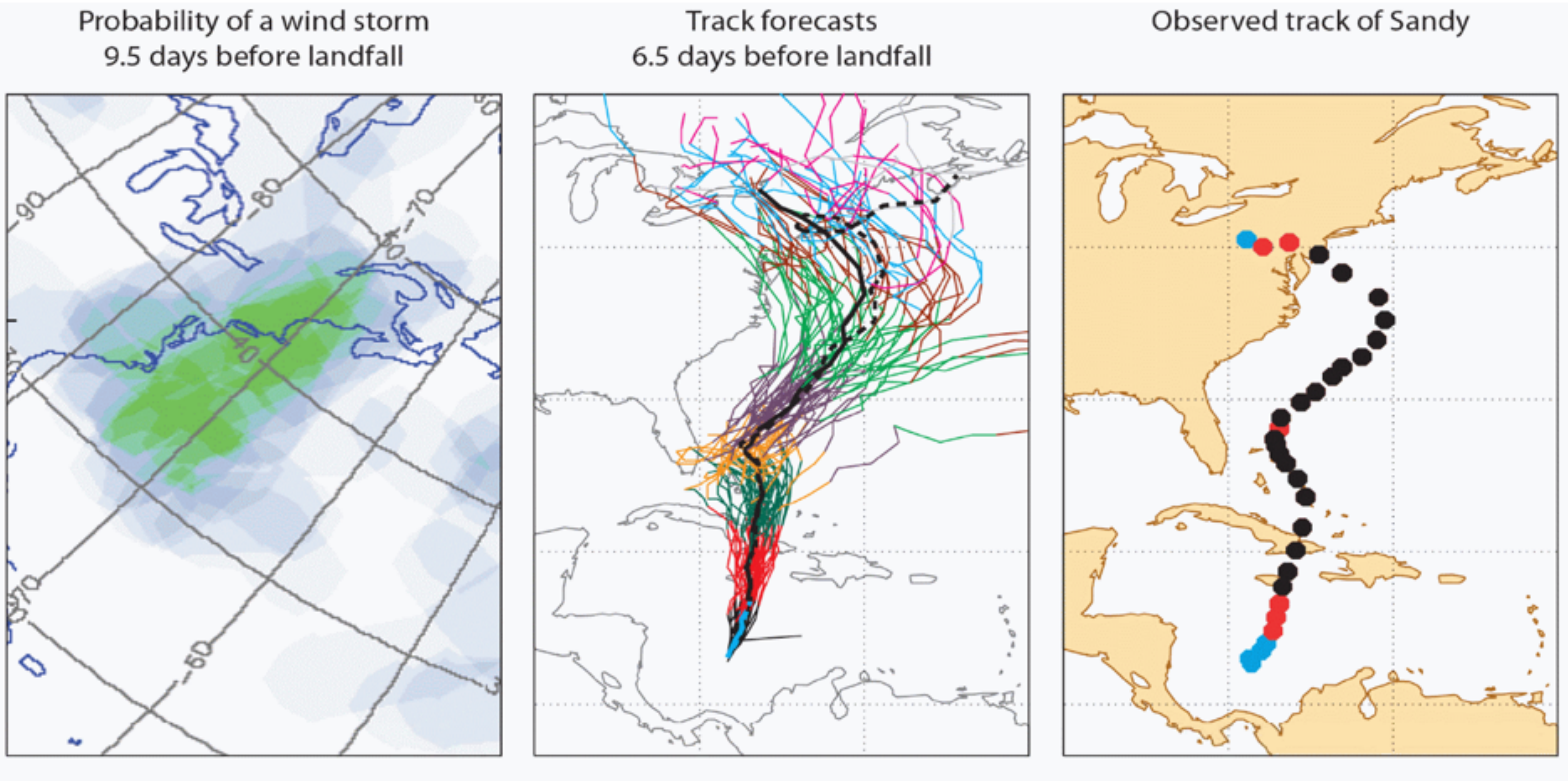
Part of the **2012 Atlantic hurricane season**



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

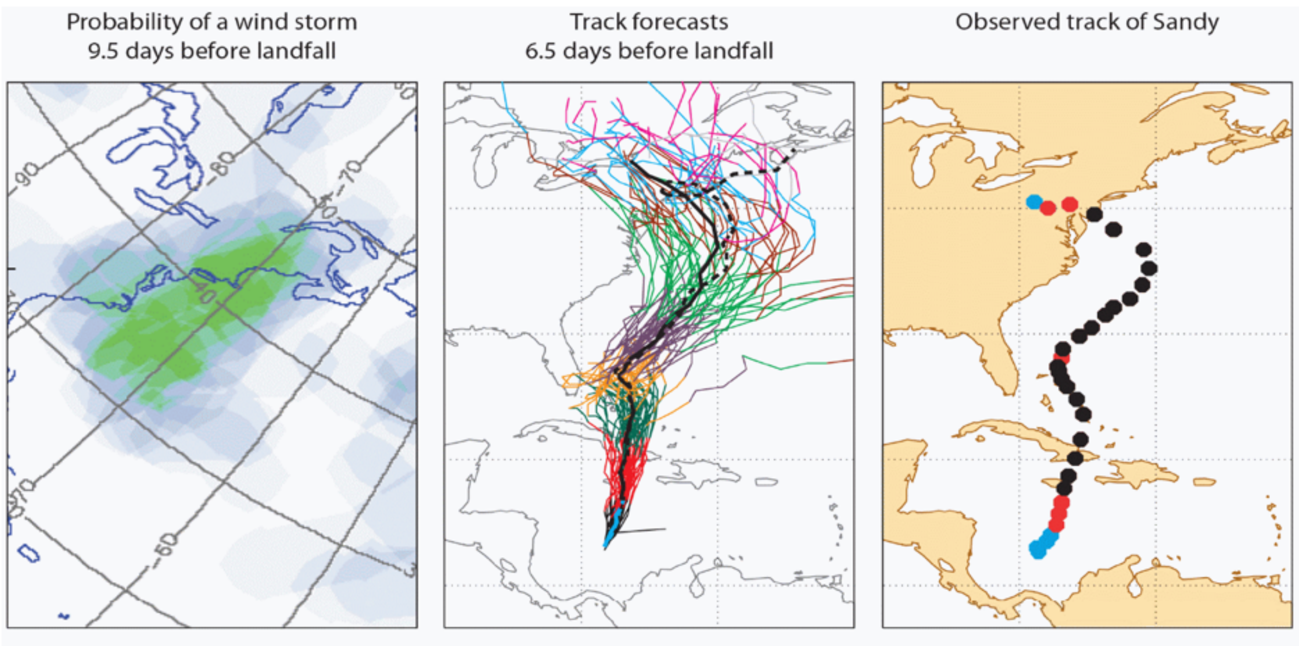


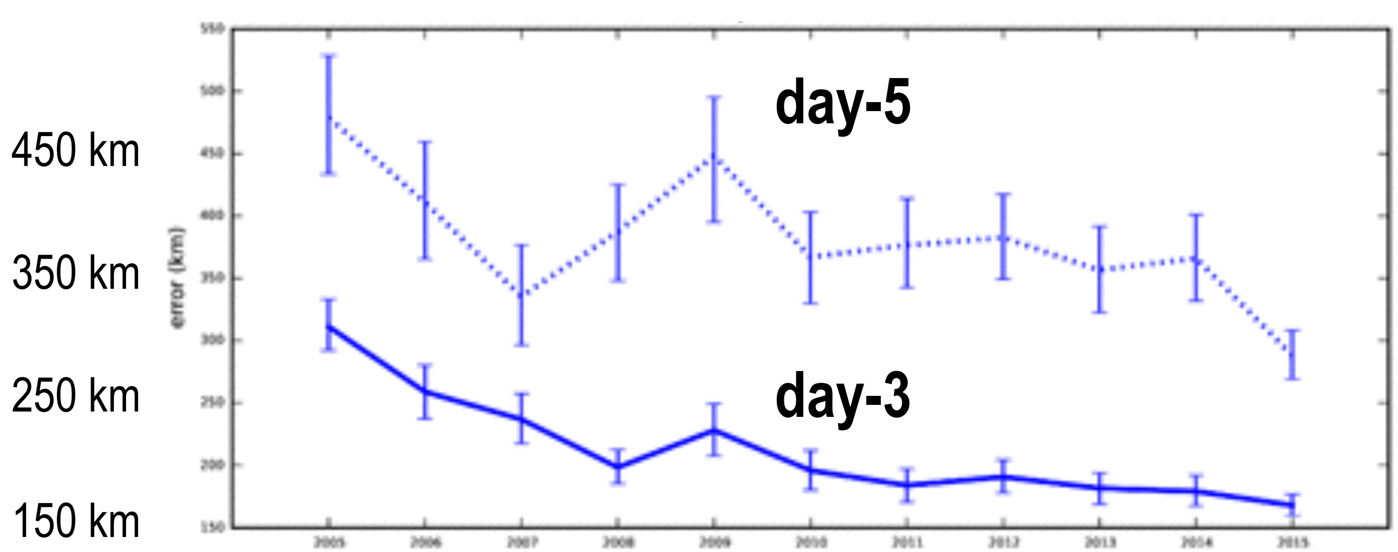
source: http://www.ecmwf.int/en/about/media-centre/news/2013/ecmwf-forecast-data-hurricane-sandy-available-researchers

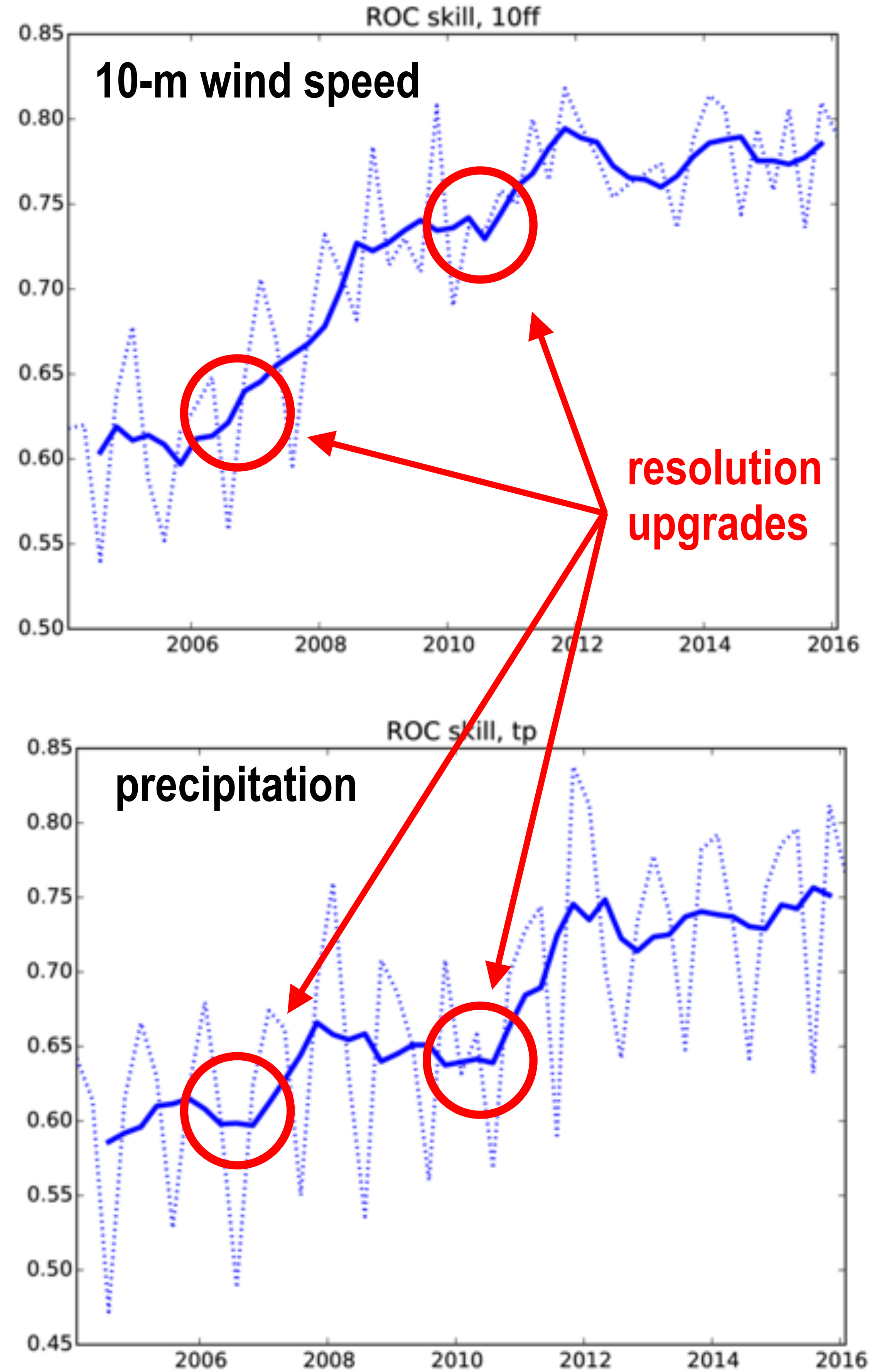

Probability of a wind storm
9.5 days before landfall
Track forecasts
6.5 days before landfall
Observed track of Sandy

Predictive skill: weather

ECMWF, source Peter Bauer







We need both, capability and throughput!

Goal: study climate extremes at km-scale resolution

Tim Palmer, Oxford

- Severe weather prediction: simulate o(1-10 days)
- Seasonal weather / climate prediction: simulate o(1 year)
  - agriculture
  - health
  - hydrological
- Multi-decadal prediction for climate adaptation: simulate o(10-100 years)
- Global prediction for informing mitigation policy: simulate o(100-1000 years)
- Geoengineering: simulate o(100-1000 years)
- Attribution of extreme weather events

HPC capability: time compression = (simulated time) / (wall clock time)

Adapting to climate change in developing countries could rise to between $280 and $500 billion p.a. by 2050

UN Adaptation Gap Report of 2016

1km-scale global simulations at exascale*?

*Exascale here is used for the timeline: DOE plans to deliver exascale supercomputers in 2023

- Today: 1km regional (refactored) models run at time compression **~100**
- If we could implement a global model with same efficiency, we can weak-scale to globe
- Beyond weak scaling we will need;
  - time compression **~1,000** for climate model in production
  - time compression **~10,000** for spin up of coupled model
- We need to accelerate the computation by **100x compared to present day simulations**

What is the baseline for today's sustained performance?

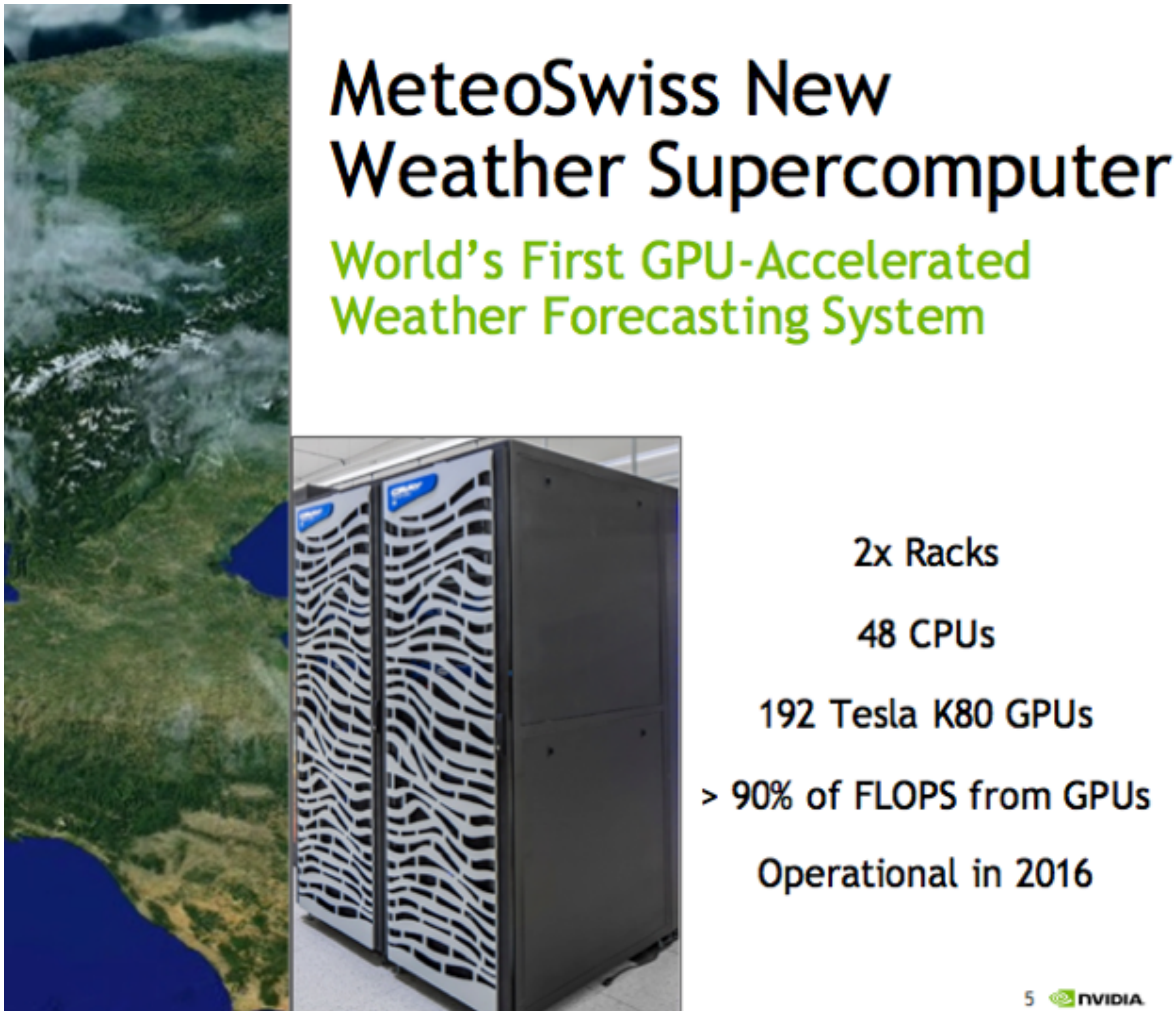
HPCwire

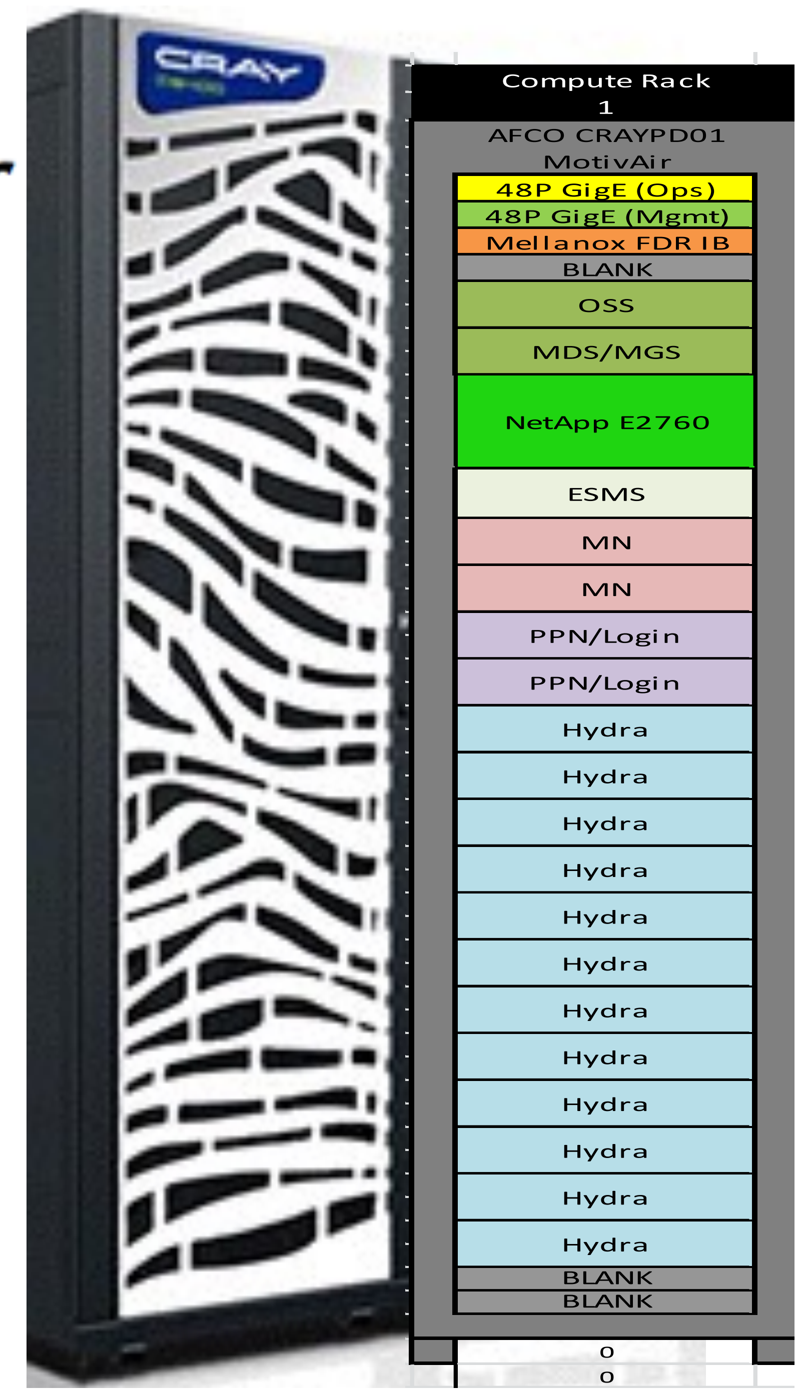
September 15, 2015

Today's Outlook: GPU-accelerated Weather Forecasting

John Russell

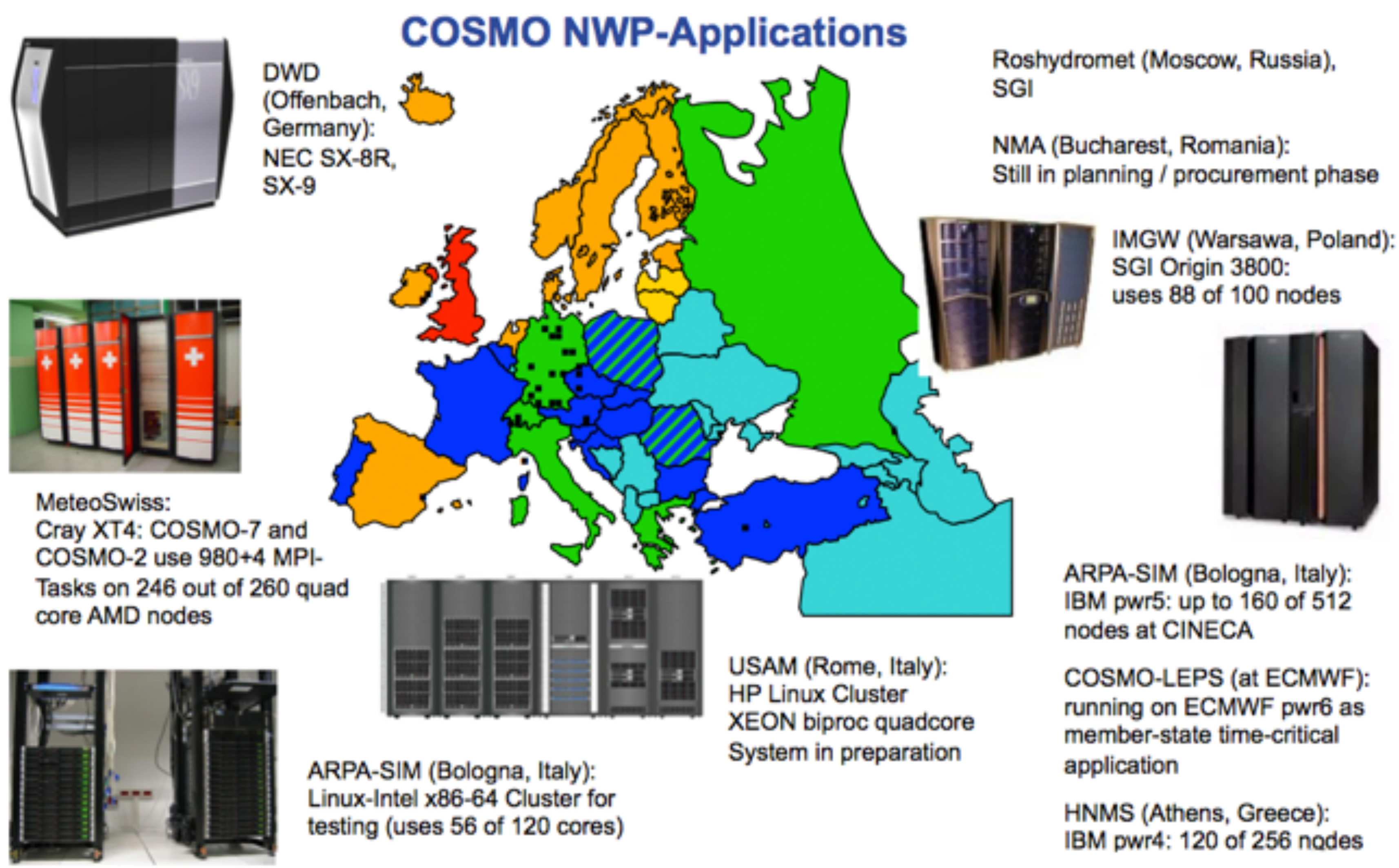
"Piz Kesch"





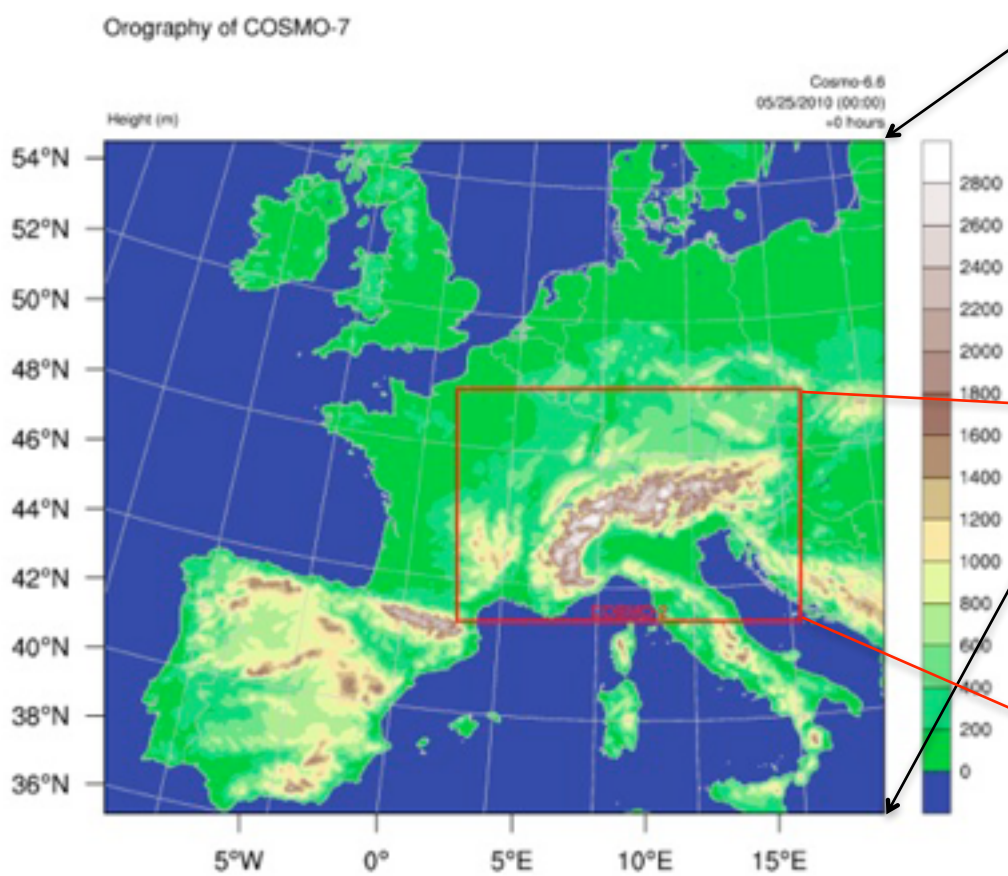
COnsortium for Small-scale Modelling (COSMO)

- Limited area model (www.cosmo-model.org)
- Used by 7 weather services and >70 research groups in academia
- Runs on many different hardware platforms!
- Very well managed consortium, in my opinion



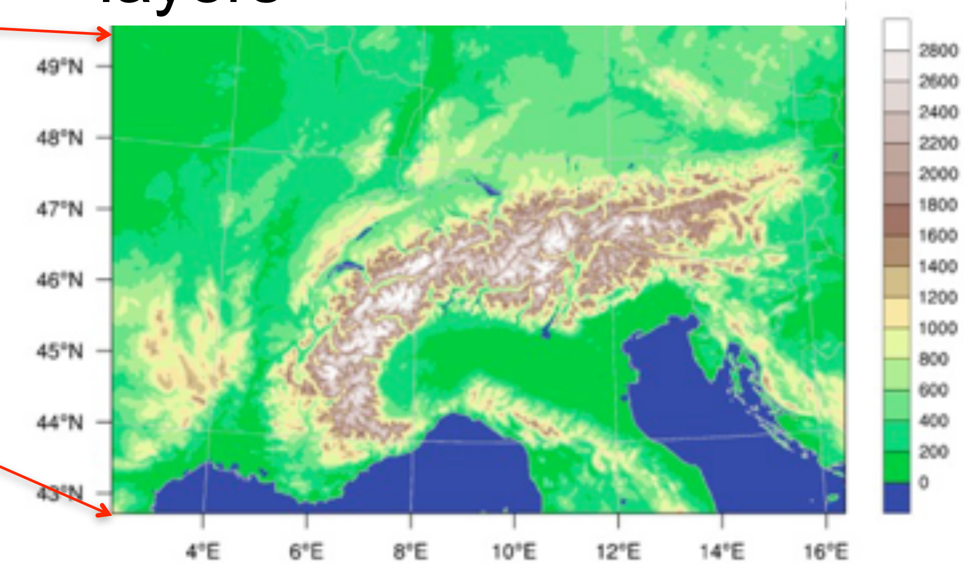
Meteo Swiss production suite until March 30, 2016

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



ECMWF
4x per day
16 km lateral grid, 91 layers

COSMO-2
8x per day 33h forecast
2.2 km lateral grid, 60 layers



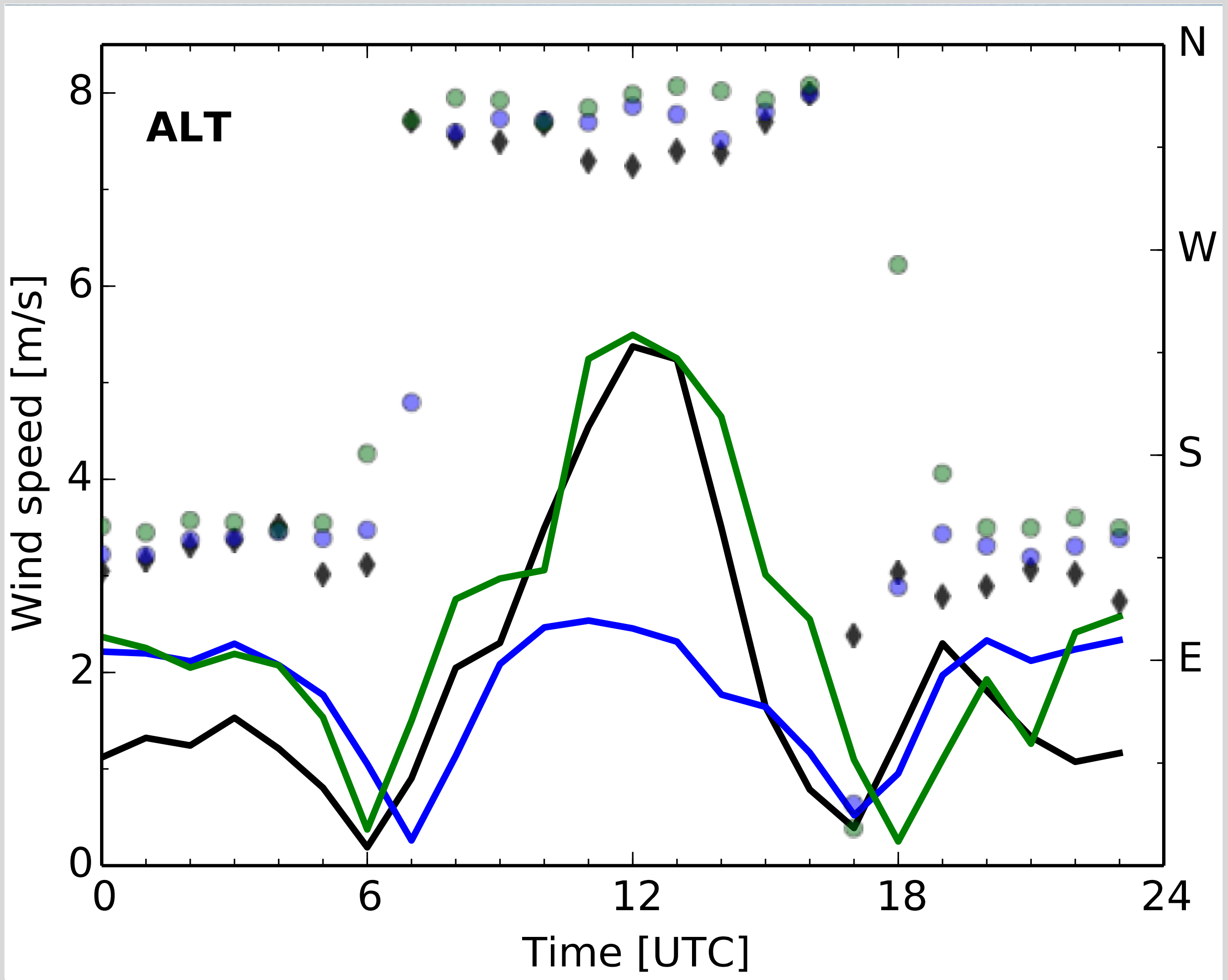
Some of the products generate from these simulations:

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

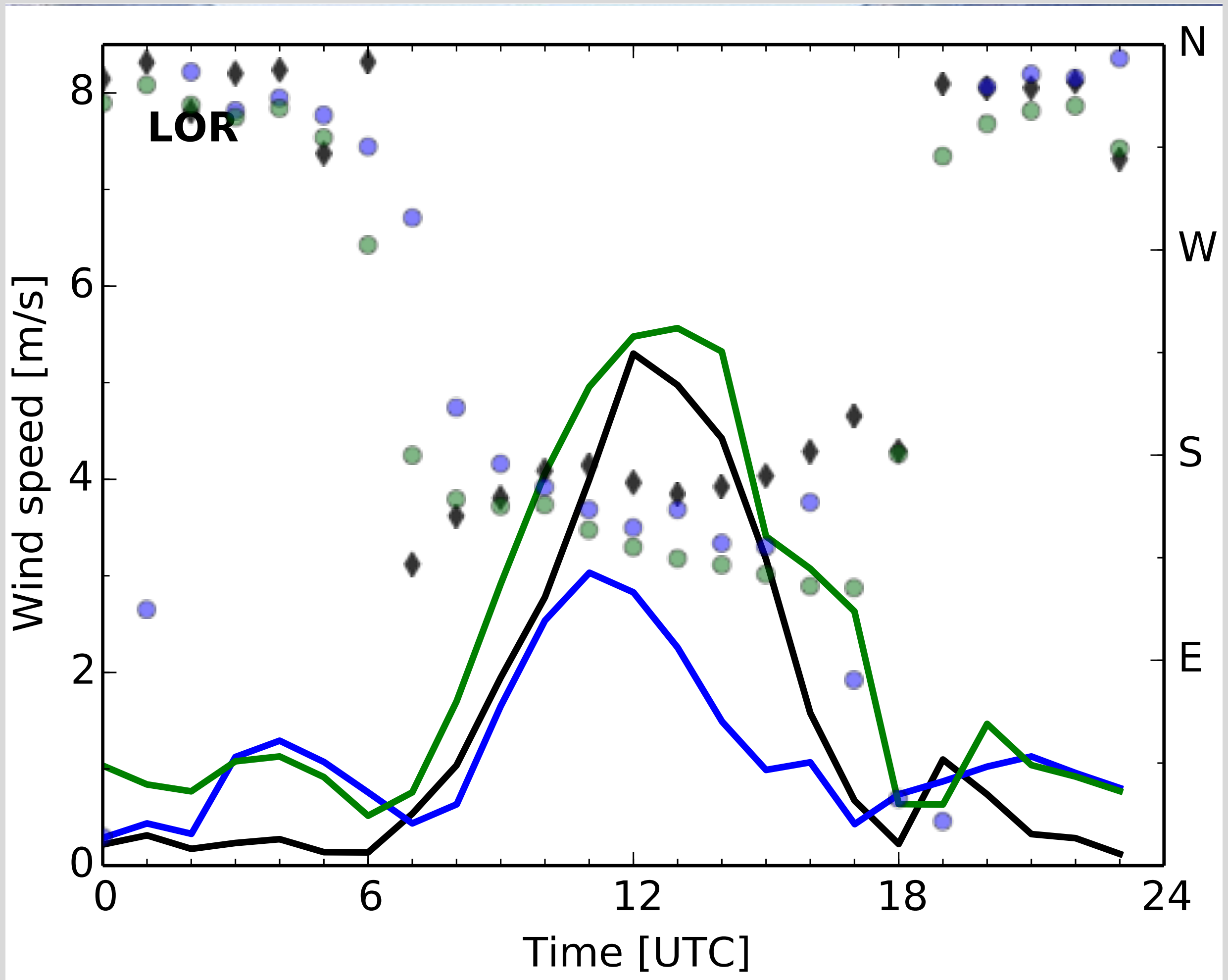
ETH zürich

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

Altdorf (Reuss valley)



Lodrino (Leventina)



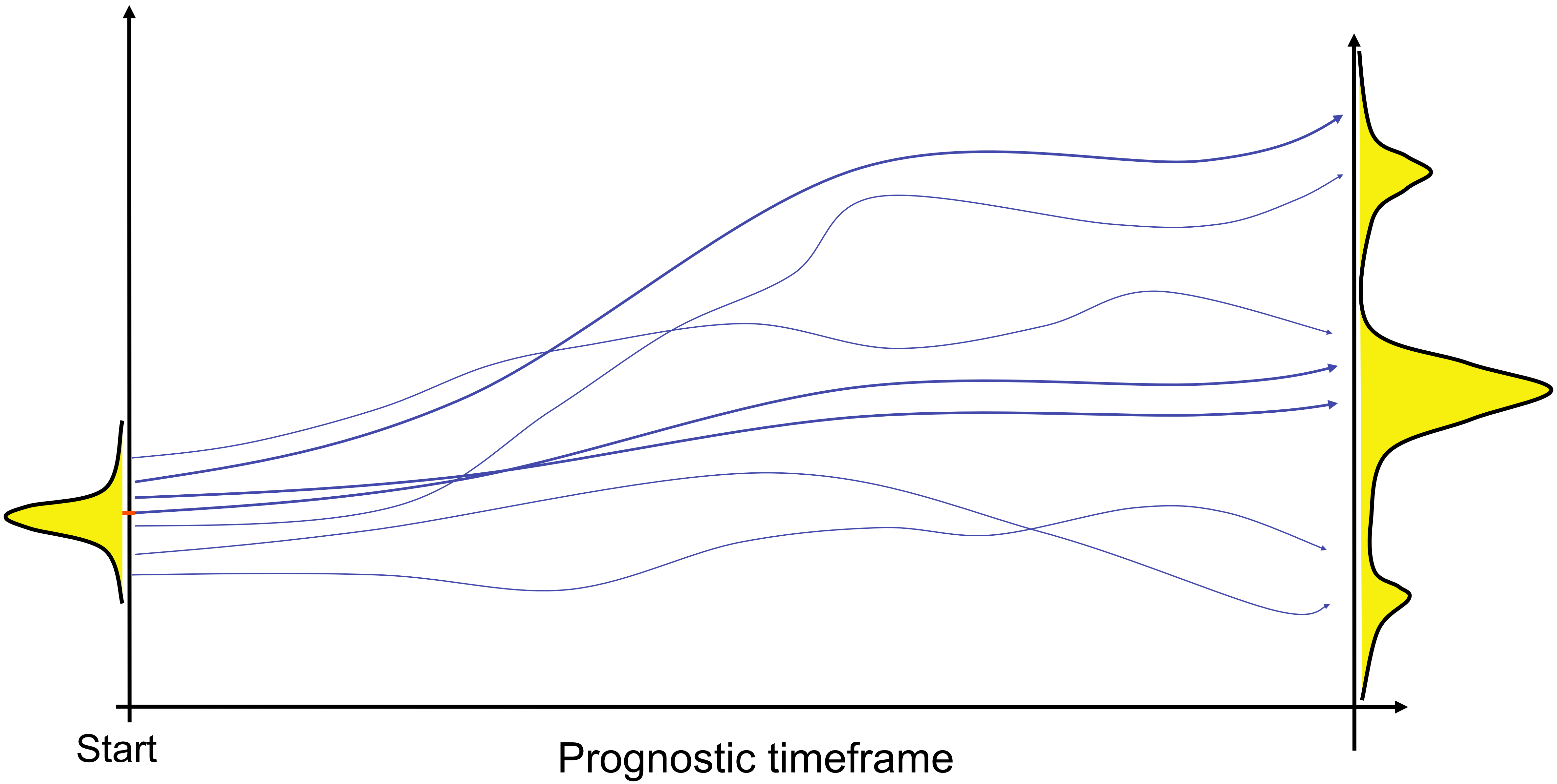
Observation Average wind speed (—) and direction (◊) **COSMO-2** **COSMO-1**

source: Oliver Fuhrer, MeteoSwiss

ETH zürich

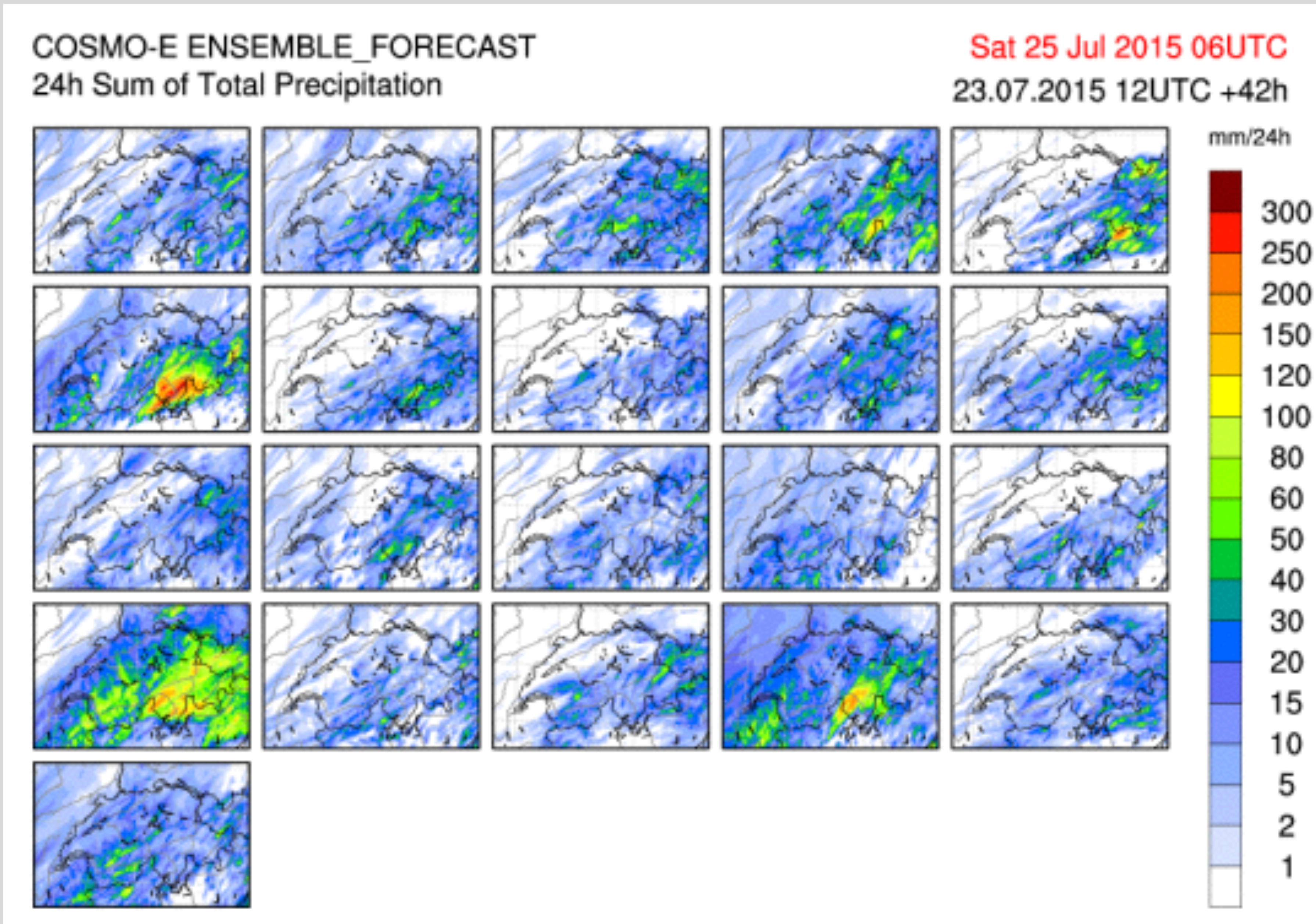
Prognostic uncertainty

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



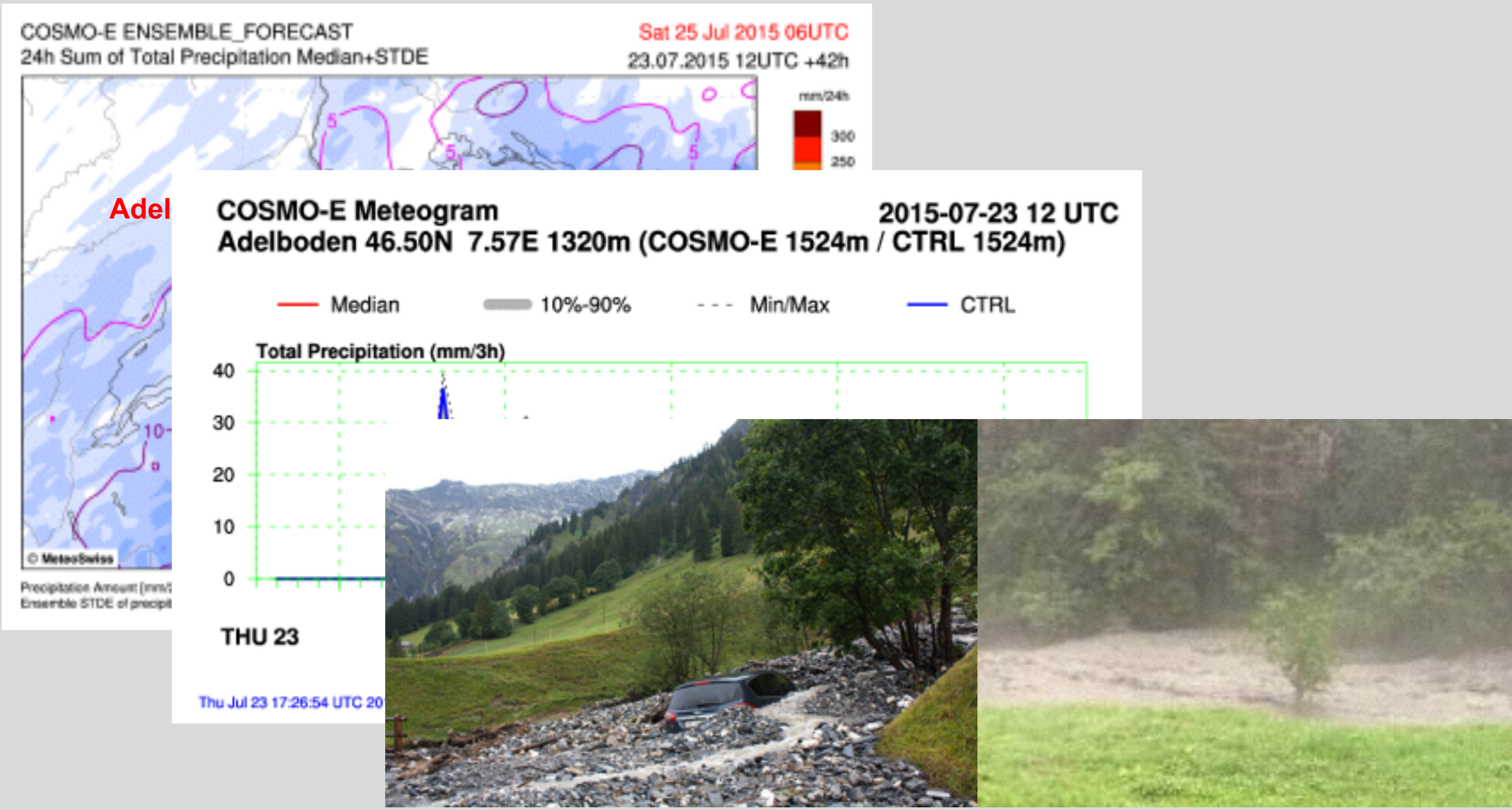
Ensemble method: compute distribution over many simulations

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

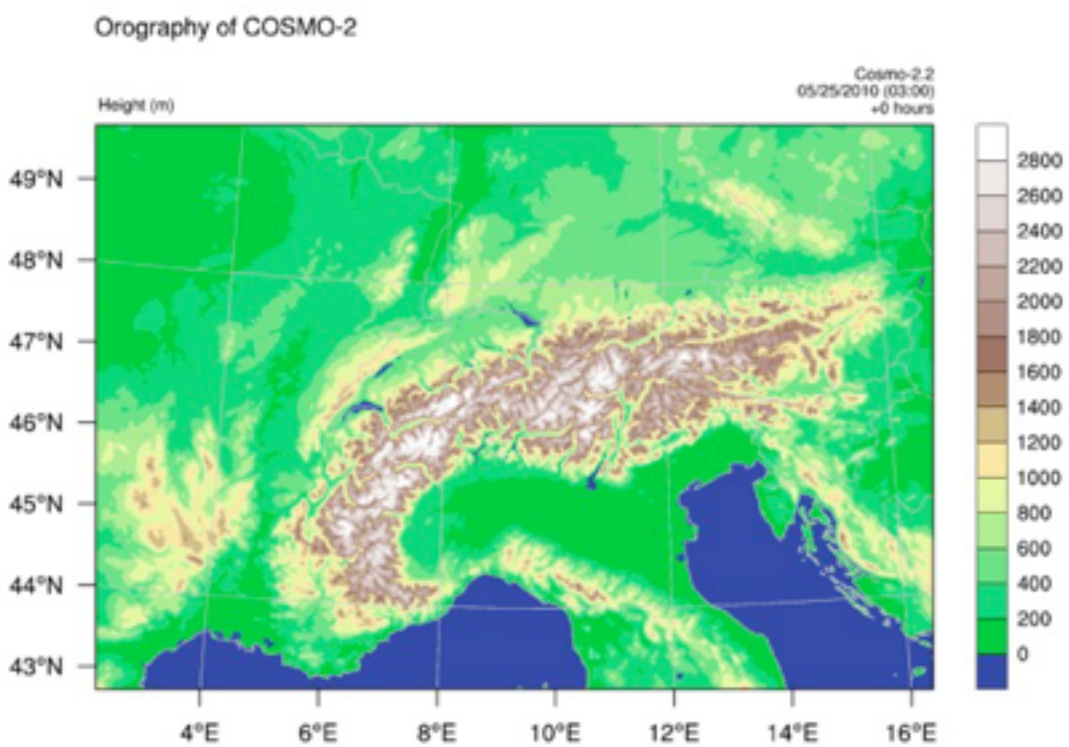
ETH zürich

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

Resource determining factors for Meteo Swiss' simulations

Operational model through March 2016

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



New model starting operation in April 2016

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

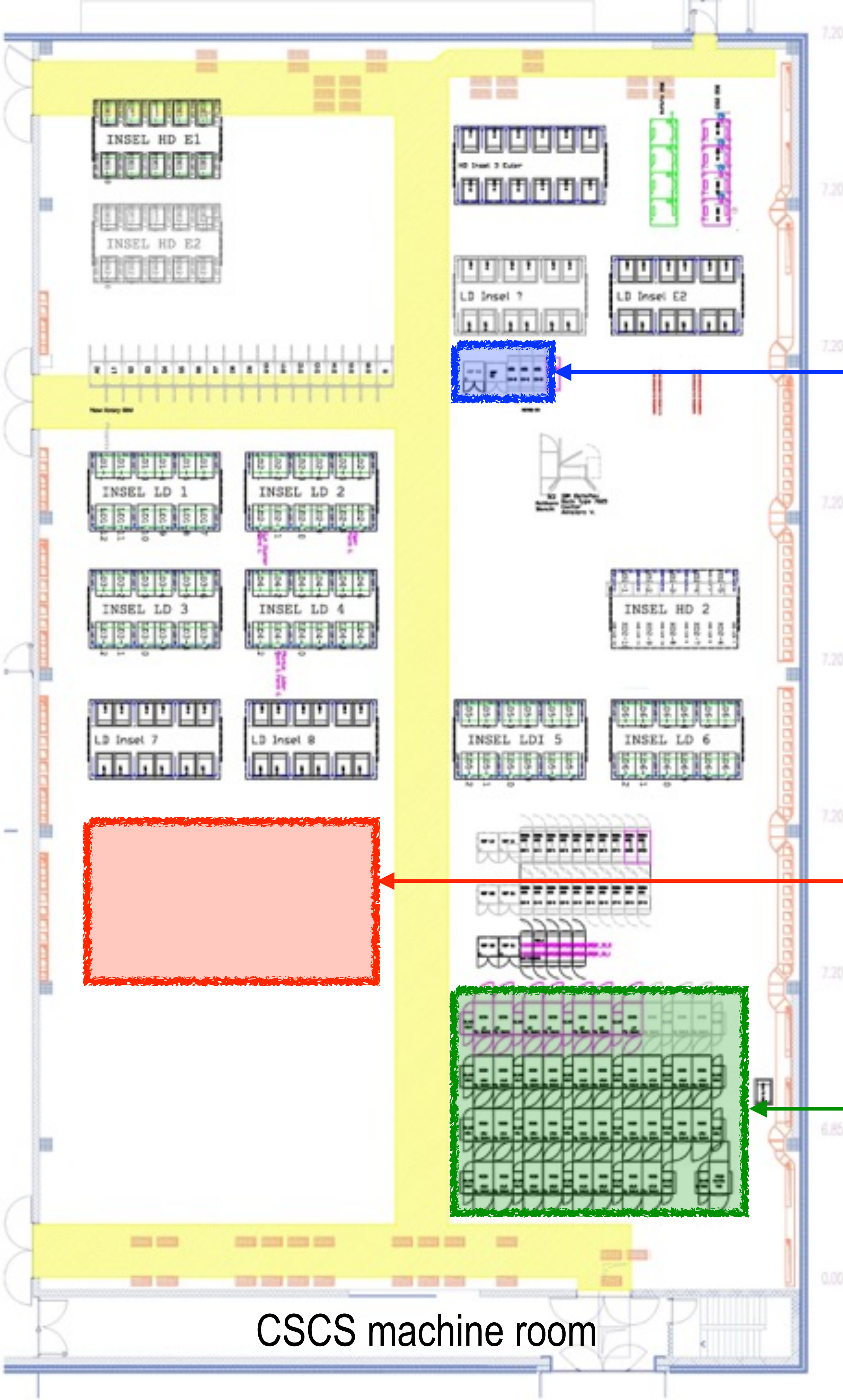
COSMO-2E: 21-member ensemble,120h forecast
in 150 min., 2x per day (**~26x** COSMO-2)

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"

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

State of the art implementation of new system for MeteoSwiss



CSCS machine room

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!

COSMO: the model Meteo Swiss uses

velocities

$$\frac{\partial u}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial\lambda} - vV_a\right\} - \dot\zeta\frac{\partial u}{\partial\zeta} - \frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial\lambda} - \frac{1}{\sqrt\gamma}\frac{\partial p_0}{\partial\lambda}\frac{\partial p'}{\partial\zeta}\right) + M_u$$

$$\frac{\partial v}{\partial t} = -\left\{\frac{1}{a}\frac{\partial E_h}{\partial\varphi} + uV_a\right\} - \dot\zeta\frac{\partial v}{\partial\zeta} - \frac{1}{\rho a}\left(\frac{\partial p'}{\partial\varphi} - \frac{1}{\sqrt\gamma}\frac{\partial p_0}{\partial\varphi}\frac{\partial p'}{\partial\zeta}\right) + M_v$$

$$\frac{\partial w}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial w}{\partial\lambda} + v\cos\varphi\frac{\partial w}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial w}{\partial\zeta} + \frac{g}{\sqrt\gamma}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial\zeta} + M_w + g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0} + \left(\frac{R_v}{R_d} - 1\right)q^v - q^l - q^f\right\}$$

pressure

$$\frac{\partial p'}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial p'}{\partial\lambda} + v\cos\varphi\frac{\partial p'}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial p'}{\partial\zeta} + g\rho_0 w - \frac{c_{pd}}{c_{vd}}pD$$

temperature

$$\frac{\partial T}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial T}{\partial\lambda} + v\cos\varphi\frac{\partial T}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial T}{\partial\zeta} - \frac{1}{\rho c_{vd}}pD + Q_T$$

water

$$\frac{\partial q^v}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^v}{\partial\lambda} + v\cos\varphi\frac{\partial q^v}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial q^v}{\partial\zeta} - (S^l + S^f) + M_{q^v}$$

$$\frac{\partial q^{l,f}}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^{l,f}}{\partial\lambda} + v\cos\varphi\frac{\partial q^{l,f}}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial q^{l,f}}{\partial\zeta} - \frac{g}{\sqrt\gamma}\frac{\rho_0}{\rho}\frac{\partial P_{l,f}}{\partial\zeta} + S^{l,f} + M_{q^{l,f}}$$

turbulence

$$\frac{\partial e_t}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial e_t}{\partial\lambda} + v\cos\varphi\frac{\partial e_t}{\partial\varphi}\right)\right\} - \dot\zeta\frac{\partial e_t}{\partial\zeta} + K_m^v\frac{g\rho_0}{\sqrt\gamma}\left\{\left(\frac{\partial u}{\partial\zeta}\right)^2 + \left(\frac{\partial v}{\partial\zeta}\right)^2\right\} + \frac{g}{\rho\theta_v}F^{\theta_v} - \frac{\sqrt2 e_t^{3/2}}{\alpha_M l} + M_{e_t}$$

Timestep implicit (sparse) explicit (RK3) implicit (sparse solver) explicit (leapfrog)

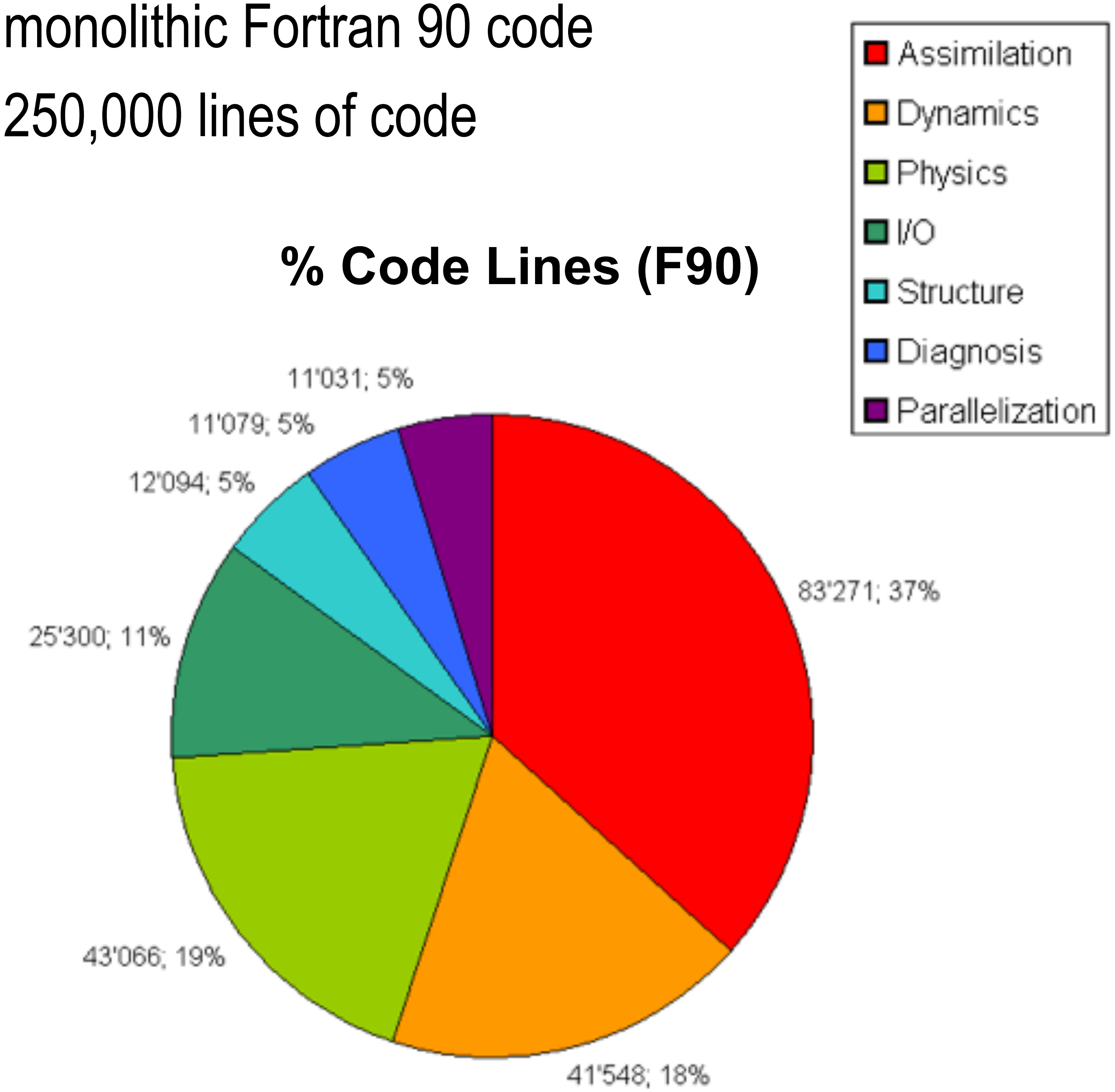
1x **physics et al. tendencies**

3x **horizontal adv.** **vertical adv.** ~10x **fast wave solver**

water adv.

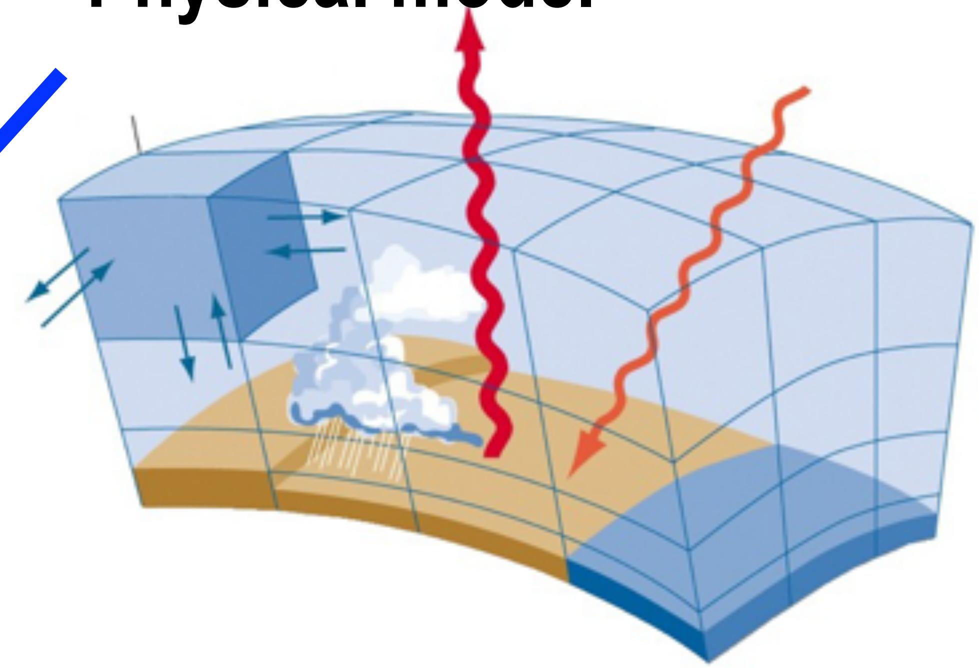
COSMO: the code behind the Meteo Swiss model

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



ETH zürich

Physical model



Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^v = -\nabla\cdot\mathbf{F}^v - (I^l + I^f)$

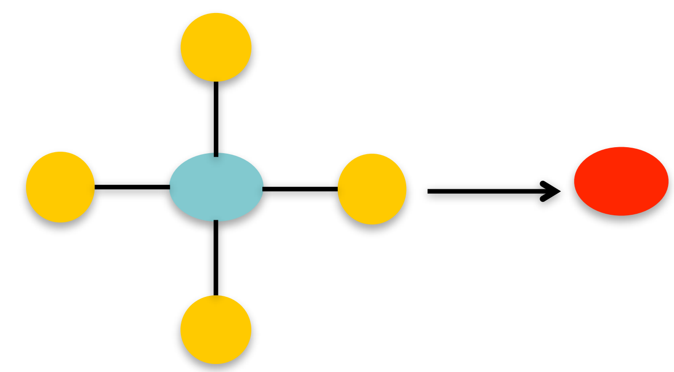
$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^v - q^l - q^f)\,T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
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);
```

Imperative code

Compilation

Computer

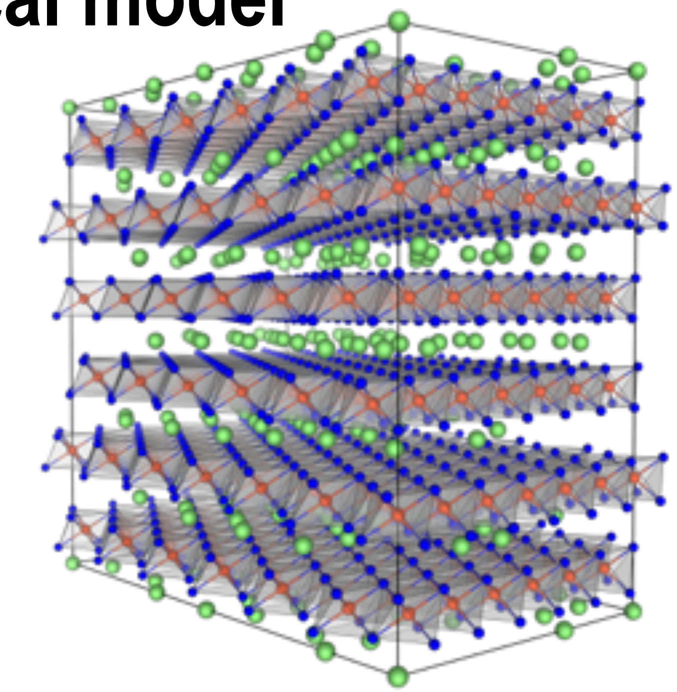
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Physical model



DFT ground state

Eigen-value problem $\hat{H}\Psi_{i\mathbf{k}}(\mathbf{r}) = \epsilon_{i\mathbf{k}}\Psi_{i\mathbf{k}}(\mathbf{r})$

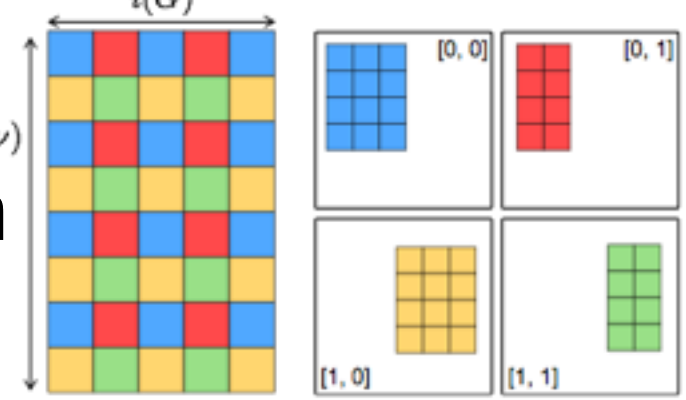
Charge density $\rho_{\sigma\sigma'}(\mathbf{r}) = \sum_{i\mathbf{k}}^{occ} \Psi^{*\sigma'}_{i\mathbf{k}}(\mathbf{r})\Psi^{\sigma}_{i\mathbf{k}}(\mathbf{r})$

Effective potential and magnetic field $v^{eff}(\mathbf{r}) = v^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$, $\mathbf{B}^{eff}(\mathbf{r}) = \mathbf{B}^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$

Mathematical description

Domain science & applied mathematics

Algorithmic description



Imperative code

Compilation

Computer

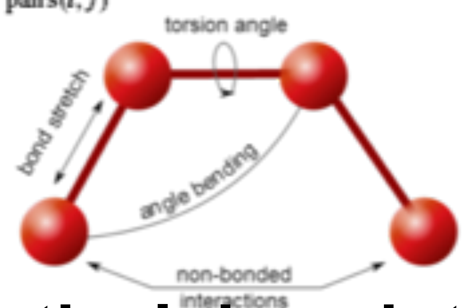
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

ETH zürich

$$V(r) = \sum_{\text{bonds}} k_b(b-b_0)^2 + \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 + \sum_{\text{dihedrals}} k_\phi(1+\cos(n\phi-\phi_0)) + \sum_{\text{impropers}} k_\psi(\psi-\psi_0)^2 + \sum_{\substack{\text{non-bonded}\\ \text{pairs}(i,j)}} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{\substack{\text{non-bonded}\\ \text{pairs}(i,j)}} \frac{q_i q_j}{\varepsilon_D r_{ij}}$$



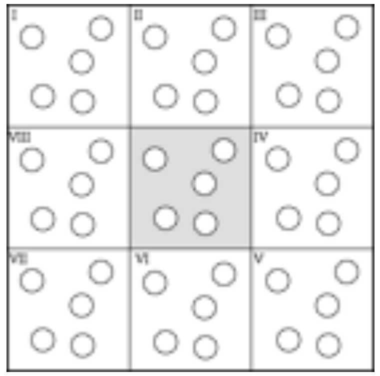
Physical model

Mathematical description

Domain science & applied mathematics

```
Cray PAT API "11 loop init mtran" in UKCA_CDAGWITHNDCL
       call pat_region_begin(11,'11 loop init mtran',pat_stat)
!
!!!        DO IMODE=1,NMODES
!!!         NDOLD(:,IMODE)=ND(:,IMODE)
!!!         DO ICP=1,NCP
!!!          MDOLD(:,IMODE,ICP)=MD(:,IMODE,ICP)
!!!          DO JMODE=1,NMODES
!!!           MTRAN(:,IMODE,JMODE,ICP)=0.0
!!!          ENDDO
!!!         ENDDO
!!!        ENDDO
!
! replace triple loops above with F90 array syntax (let
compiler decide)
       NDOLD=ND
       MDOLD=MD
       MTRAN=0.0
       call pat_region_end(11,pat_stat)
```

Algorithmic description



Imperative code

Compilation

Computer

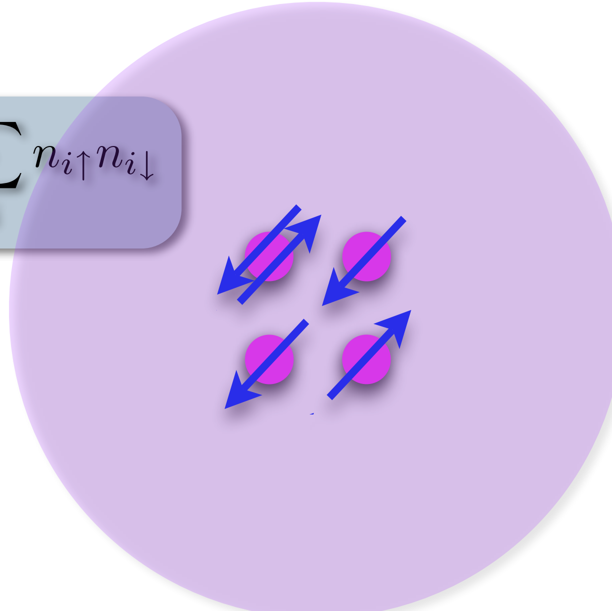
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

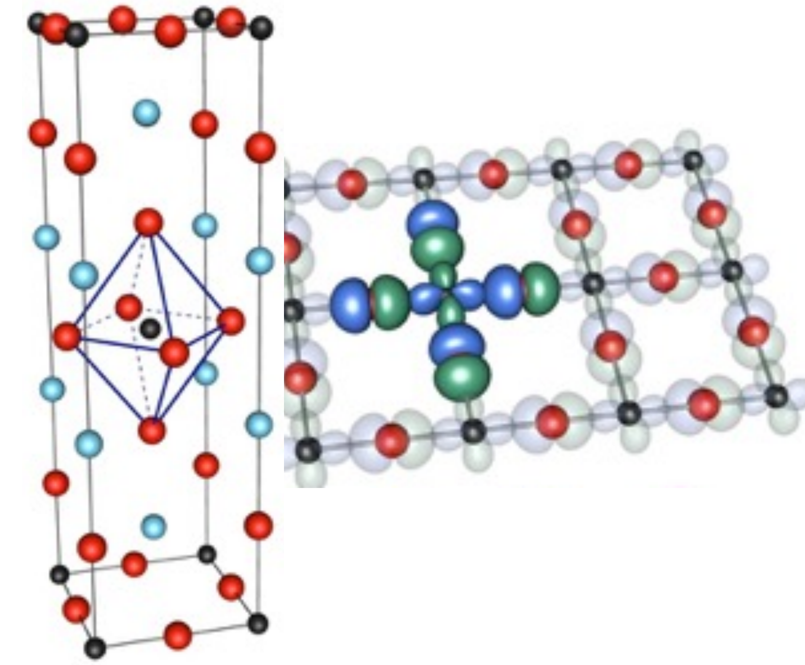
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

$$\mathcal{H} = -t \sum_{\langle ij\rangle,\sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$



Physical model

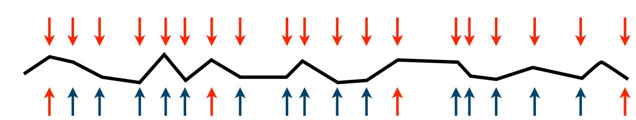


Mathematical description

Domain science & applied mathematics

Algorithmic description

$$\mathbf{G}_c(\{s_i,l\}_{k+1}) = \mathbf{G}_c(\{s_i,l\}_k) + \mathbf{a}_k \times \mathbf{b}_k^t$$



$$\mathbf{G}_c(\{s_i,l\}_{k+1}) = \mathbf{G}_c(\{s_i,l\}_0) + [\mathbf{a}_0|\mathbf{a}_1|...|\mathbf{a}_k] \times [\mathbf{b}_0|\mathbf{b}_1|...|\mathbf{b}_k]^t$$

Imperative code

Compilation

Computer

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Physical model

Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

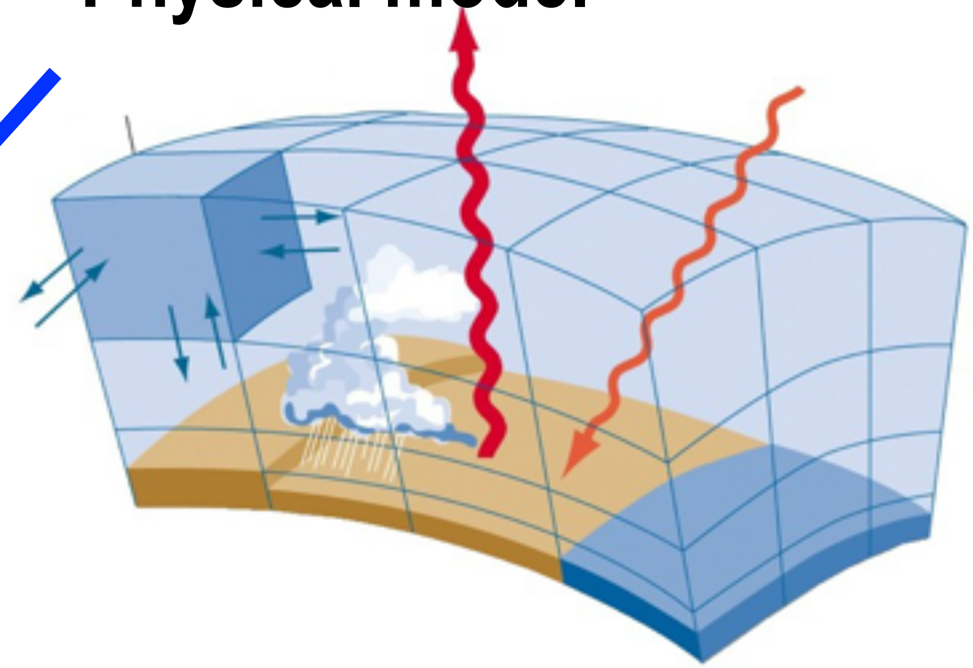
Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\,\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^v = -\nabla\cdot\mathbf{F}^v - (I^l + I^f)$

$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

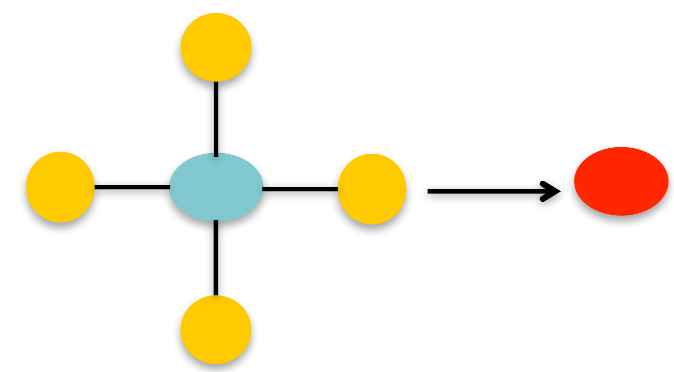
Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^v - q^l - q^f)\,T]^{-1}$



Mathematical description

Domain science & applied mathematics

Algorithmic description



```
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);
```

Imperative code

Compilation

Computer

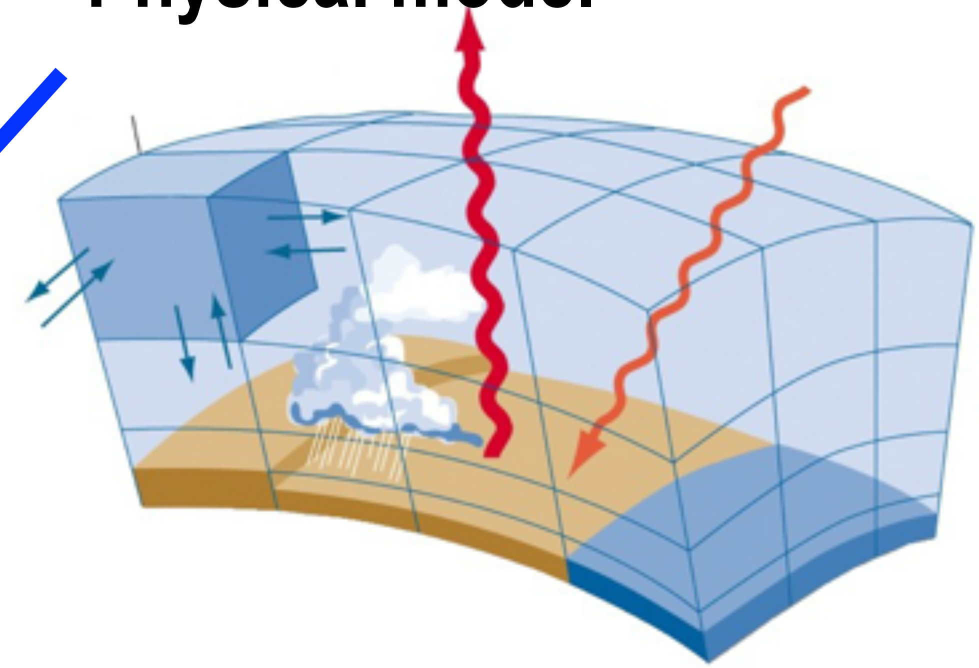
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Physical model



$$\text{Wind } \rho\dot{\mathbf{v}} = -\nabla p + \rho \mathbf{g} - 2\Omega \times (\rho \mathbf{v}) + \mathbf{F}$$

$$\text{Pressure } \dot{p} = -(c_{pd}/c_{vd})\, p \nabla \cdot \mathbf{v} + (c_{pd}/c_{vd} - 1) Q_h$$

$$\text{Temperature } \rho c_{pd} \dot{T} = \dot{p} + Q_h$$

$$\text{Water } \rho \dot{q}^{v} = -\nabla \cdot \mathbf{F}^{v} - (I^{l} + I^{f})$$

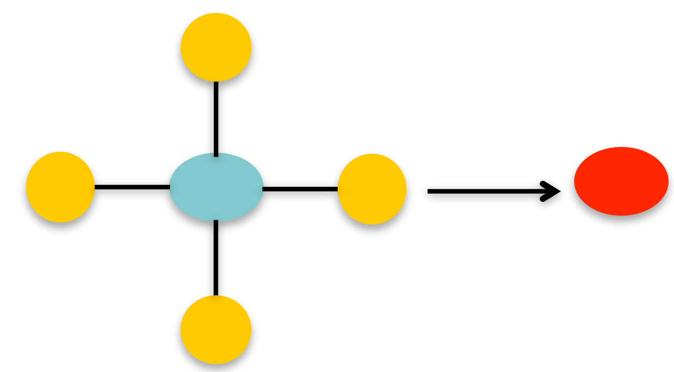
$$\rho \dot{q}^{l,f} = \nabla \cdot (\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$$

$$\text{Density } \rho = p[R_d (1 + (R_v/R_d - 1) q^{v} - q^{l} - q^{f}) T]^{-1}$$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
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);
```

Imperative code

Compilation

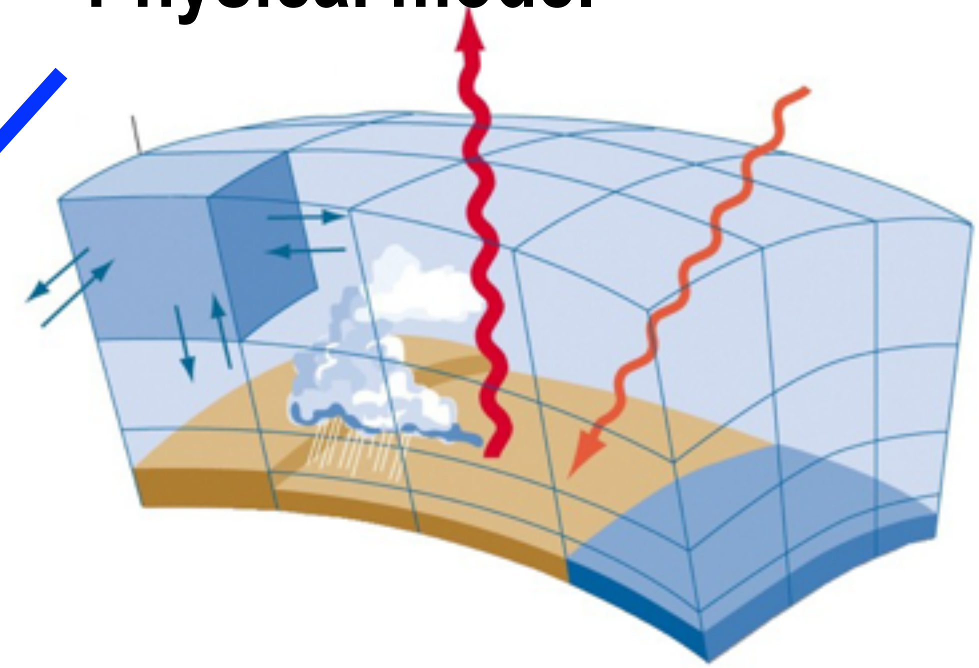
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Physical model



Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\dot{T} = \dot{p}+Q_h$

Water $\rho\dot{q}^{v} = -\nabla\cdot\mathbf{F}^{v} - (I^{l}+I^{f})$

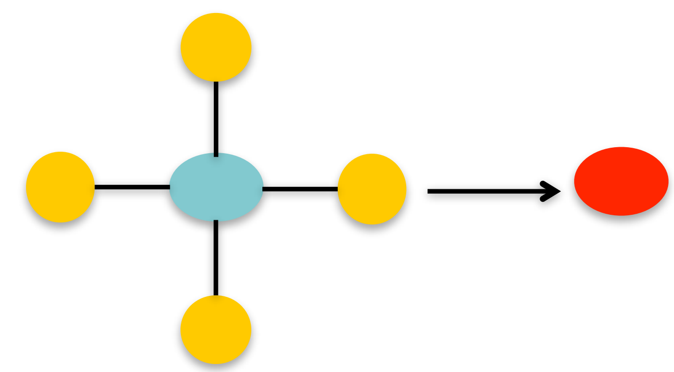
$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^{v}-q^{l}-q^{f})\,T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
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);
```

Imperative code

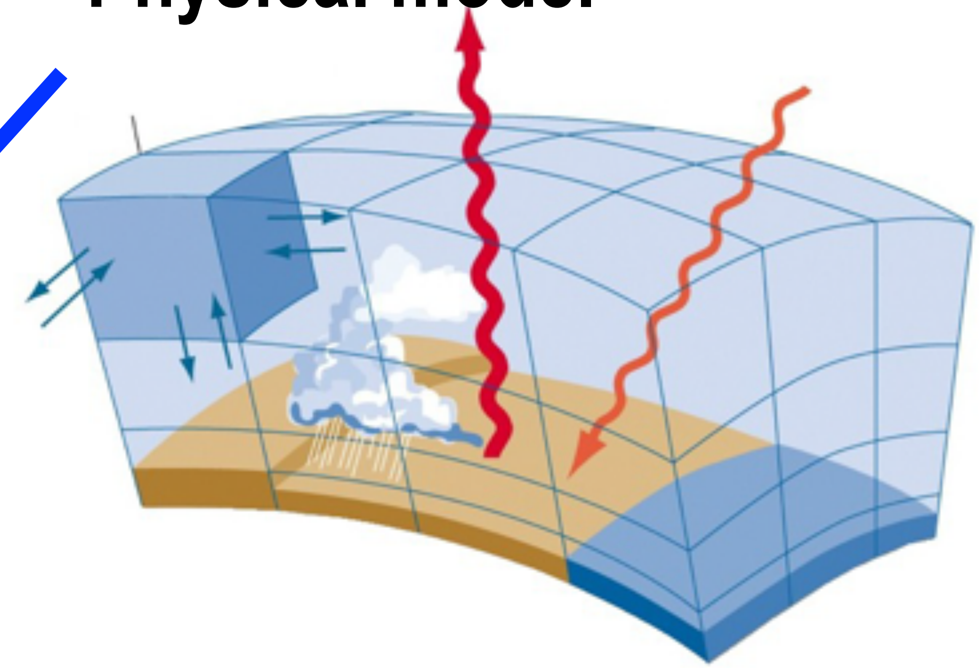
Compilation

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Physical model



Wind $\rho \dot{\mathbf{v}} = -\nabla p + \rho \mathbf{g} - 2\Omega \times (\rho \mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p \nabla \cdot \mathbf{v} + (c_{pd}/c_{vd} - 1) Q_h$

Temperature $\rho c_{pd} \dot{T} = \dot{p} + Q_h$

Water $\rho \dot{q}^{v} = -\nabla \cdot \mathbf{F}^{v} - (I^{l} + I^{f})$

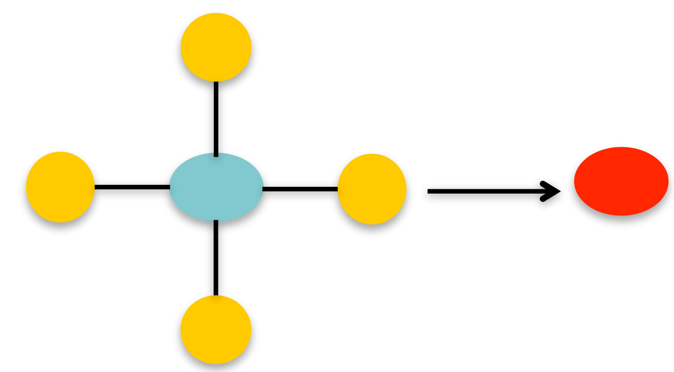
$\rho \dot{q}^{l,f} = \nabla \cdot (\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1 + (R_v/R_d - 1)\, q^{v} - q^{l} - q^{f})\, T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
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);
```

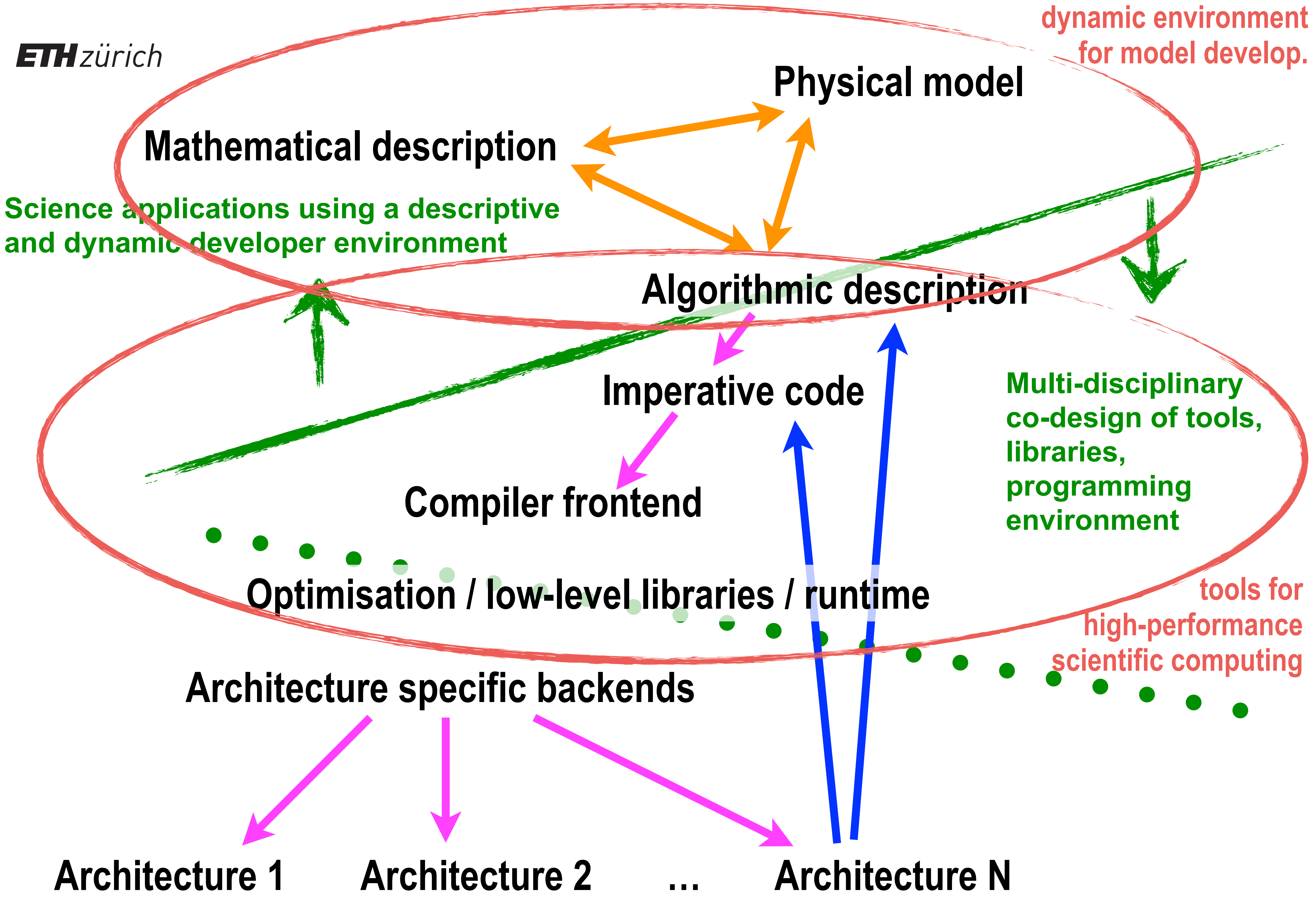
Imperative code

Compilation

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

ETH zürich

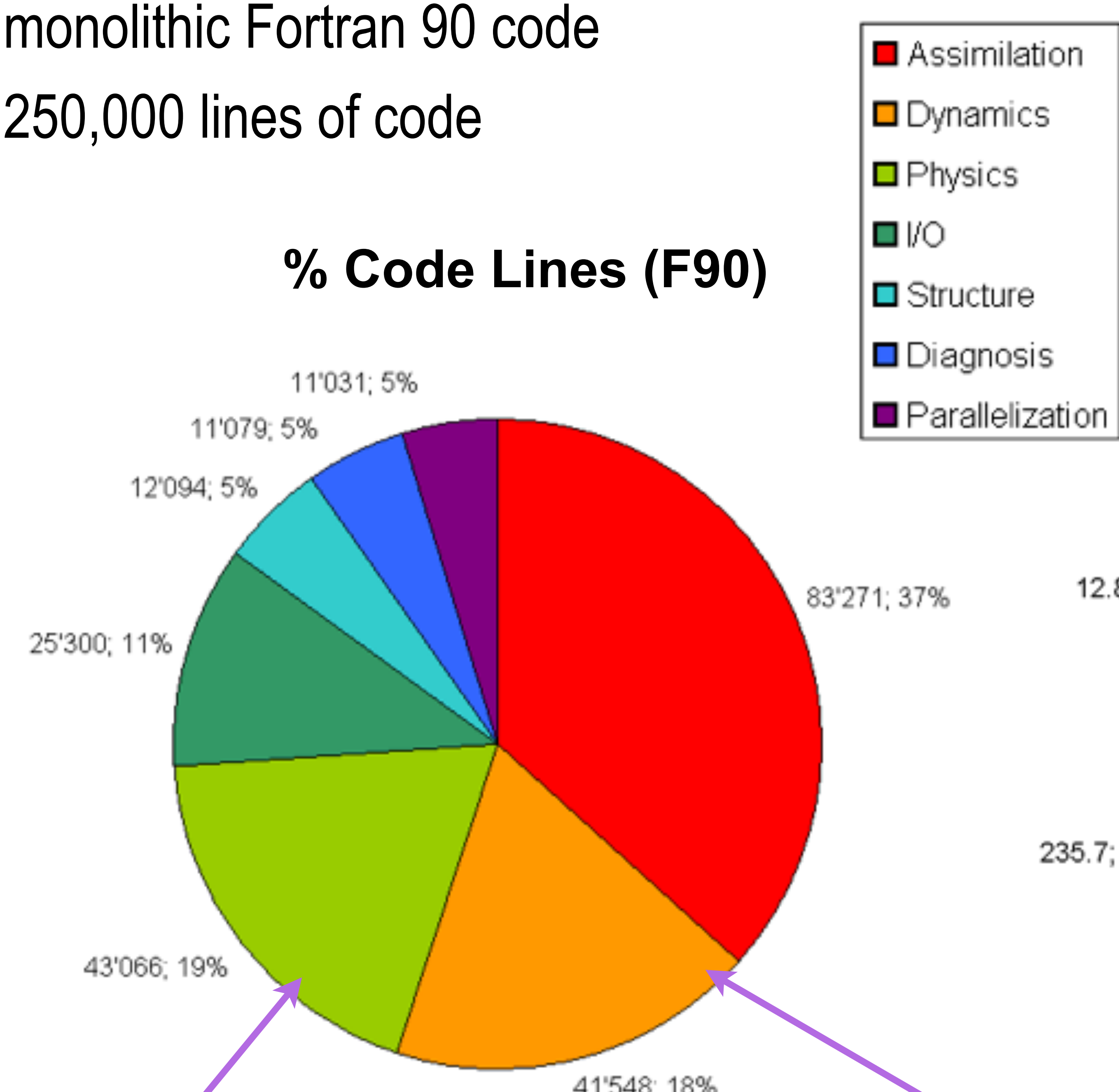


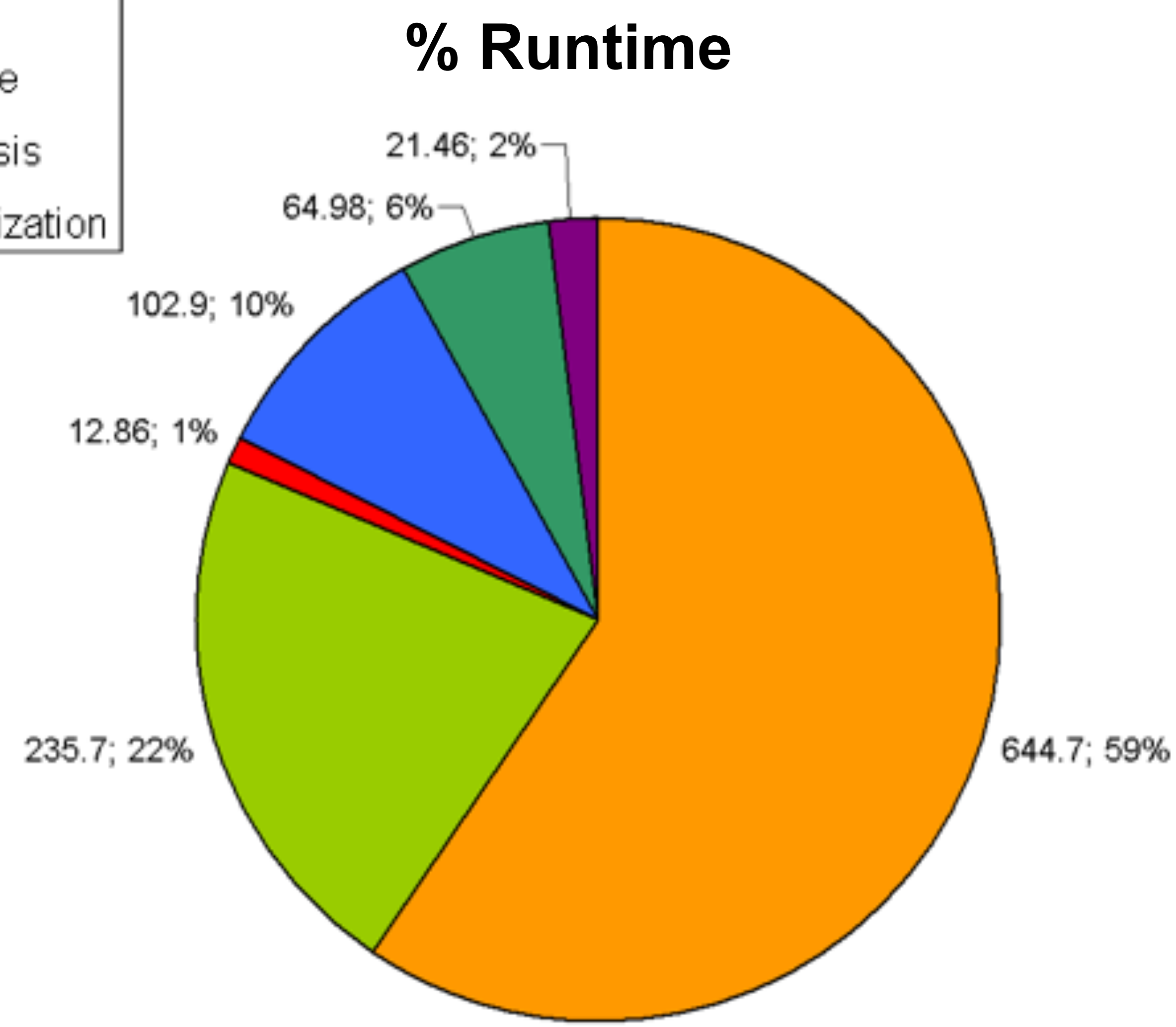
Schulthess, Nature Physics, vol 11, 369-373 (2015)

COSMO: a legacy code migration project

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

Runtime based 2 km production model of MeteoSwiss



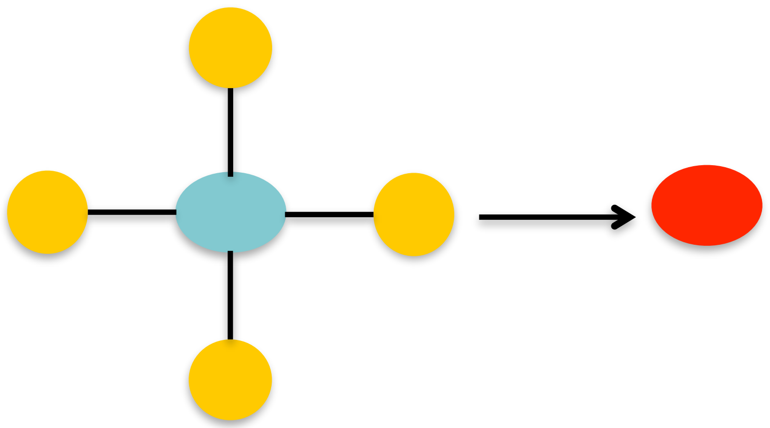


Original code (with OpenACC for GPU)

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

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
```

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
```

ETH zürich

```
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

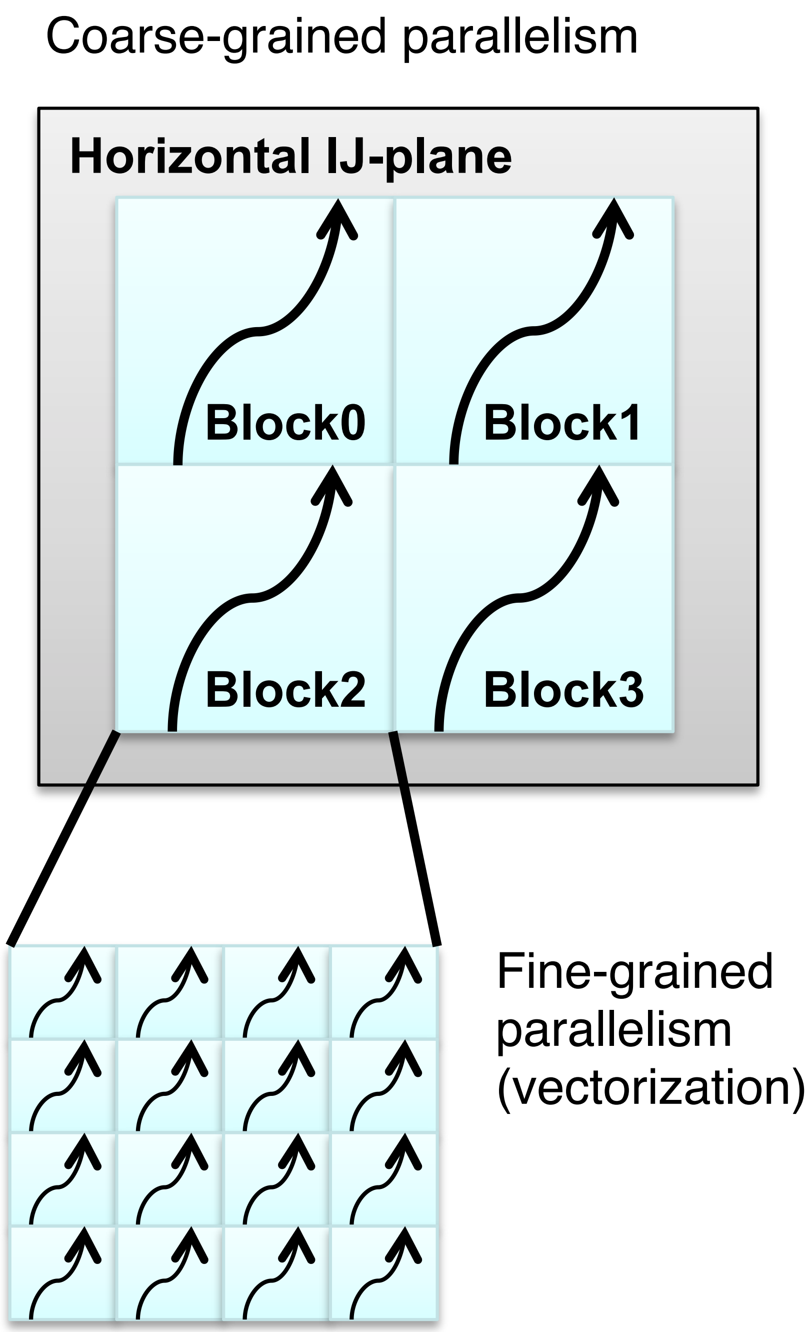
```
enum { data, lap };

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

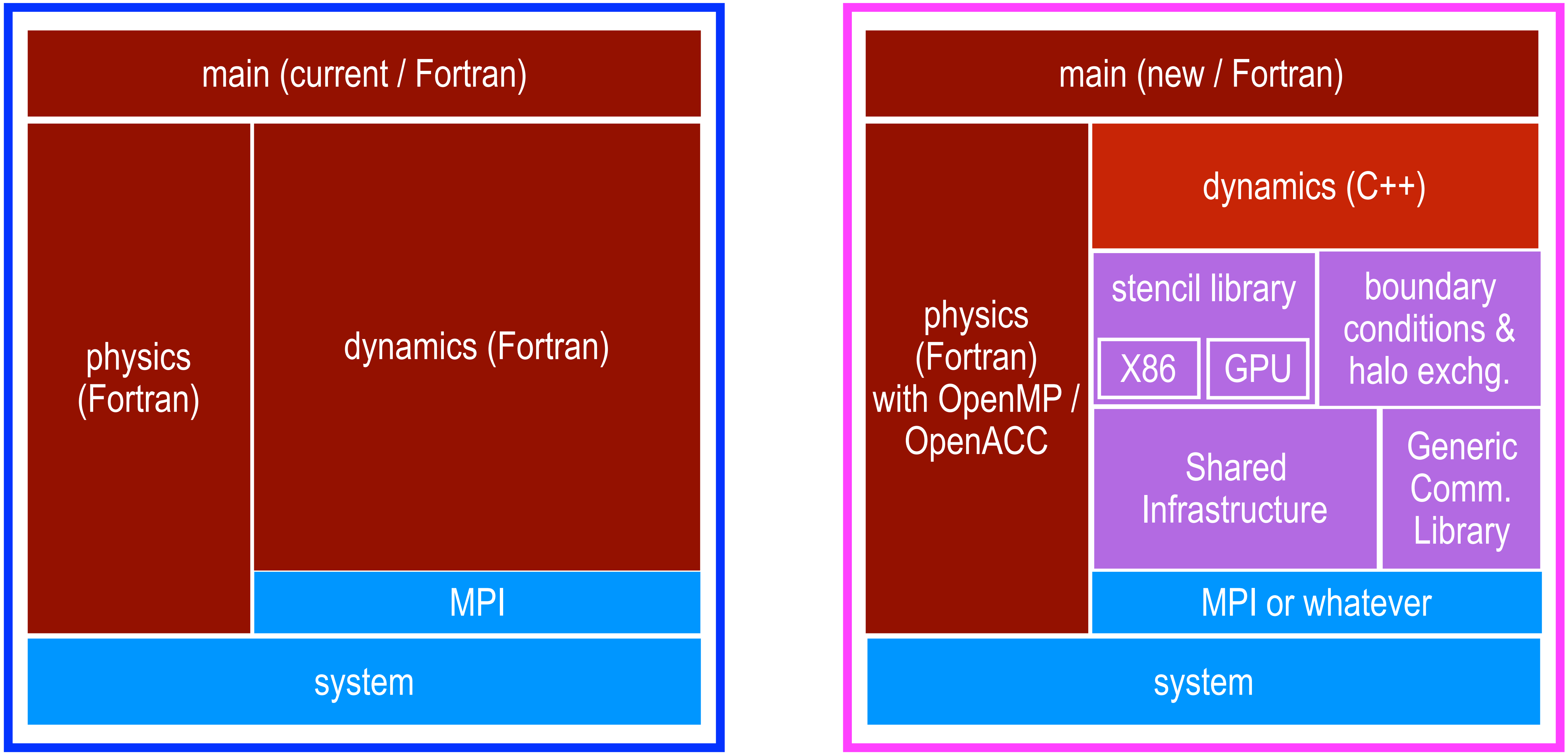
```
IJKRealField lapfield, datafield;
Stencil stencil;

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

stencil.Apply();
```

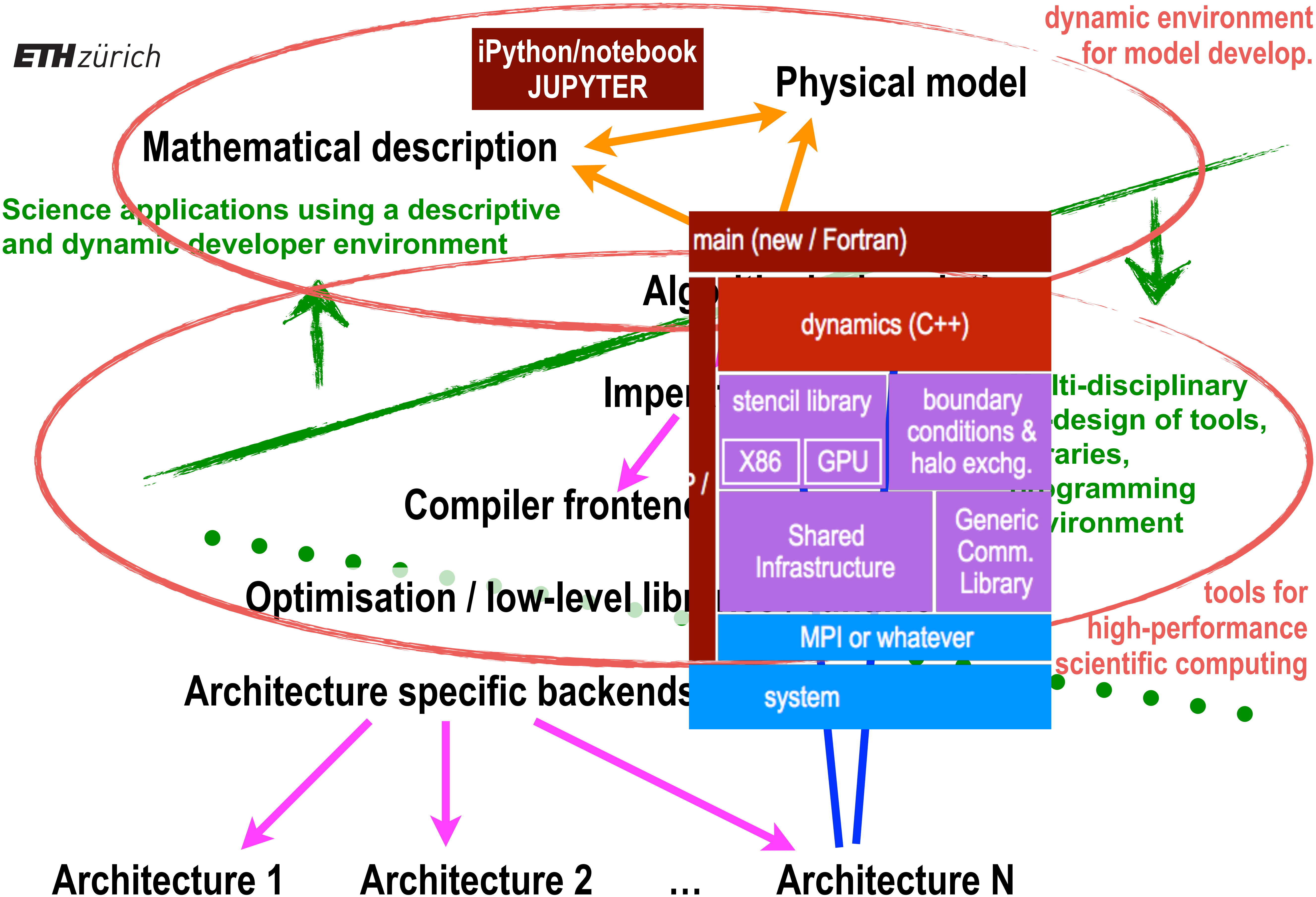
Architecture dependent backend

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



COSMO: old and new (refactored) code





Schulthess, Nature Physics, vol 11, 369-373 (2015)

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

What we compare to establish the baseline

- Three machine types
  - Cray XE6 with AMD Barcelona – state of the art in 2012
  - Cray XC40 with Intel Xeon (Haswell) – state of the art in 2015
  - Cray CS Storm with Intel Xeon (Haswell) and NVIDIA K80 GPU – state of the art in 2015
- Two implementations of the COSMO model
  - Standard F90 with MPI – used by German Weather Service and others
  - Refactored, hybrid F90 + C++ with MPI & CUDA / OpenMP – used by MeteoSwiss

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)

- Past production system installed in 2012
- New Piz Kesch/Escha installed in 2015
  - Processor performance (X86) **2.8x** ← Moore's Law
  - Algorithms & system utilisation **2.8x**
  - General software performance **1.7x** ← Software refactoring
  - 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 & ~30% reduction in power consumption

Albis & Lema (in production through 3/2016) New system: Kesch & Escha

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)

- Past production system installed in 2012
- New Piz Kesch/Escha installed in 2015
  - Processor performance **2.8x**
  - Algorithms & system utilisation **2.8x**
  - General software performance **1.7x**
  - Port to GPU architecture **2.3x**
  - 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

(Algorithms & system utilisation, general software performance and port to GPU architecture together: = 11x)

So what is the baseline for exascale?

The state-of the art implementation of COSMO running at DWD (Deutscher Wetterdienst) on multi-core hardware.

~10x

The refactored version of COSMO running at MeteoSwiss on multi-core or GPU accelerated hardware.

1km-scale global simulations at exascale*?

*Exascale here is used for the timeline: DOE plans to deliver exascale supercomputers in 2023

- Today: 1km regional (refactored) models run at time compression **~100**
- If we could implement a global model with same efficiency, we can weak-scale to globe
- Beyond weak scaling we will need;
  - time compression **~1,000** for climate model in production
  - time compression **~10,000** for spin up of coupled model
- We need to accelerate the computation by **100x compared to present day simulations**
- Example of COSMO, ICON (assuming the latter is as efficient at the former)
  - Maybe speed up another factor 2 in strong scaling with current algorithms
  - Expected improvements in hardware (2019) ~3x
  - Maybe there is another factor 2 in hardware by early 2020s
- In the best of cases will need at least another factor 10 from somewhere else
  - consider methods / algorithms
  - co-design a more appropriate computing system?

Conclusion

Is it realistic to have the exascale systems we need in 2020 (China) or 2023 (USA/Japan)?

The answer depends on what you 2016 baseline is!

But for sure we will need more than exascale to solve our real problems (e.g. climate and meteorology)

Outside the box solutions are needed

ETH zürich

PASC17

Platform for Advanced Scientific Computing Conference

Lugano Switzerland | 26-28 June 2017

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre