High-Performance C++ in Weather Prediction

Challenges, Achievements and Future Models

Pascal Spörri[1], Andrea Arteaga[2], Oliver Fuhrer[2], Tobias Gysi[3], Carlos Osuna[2], Thomas C. Schulthess[4,5,6]

[1]Center for Climate Systems Modeling, ETH Zürich; [2]Federal Office of Meteorology and Climatology, MeteoSwiss; [3]Department of Computer Science, ETH Zürich; [4]Swiss National Supercomputing Centre, ETH Zürich; [5]Institute for Theoretical Physics, ETH Zürich; [6]Computer Science and Mathematics Division, Oak Ridge National Laboratory

Production System

Piz Escha **Piz Kesch**

2x Racks:
1x Production (Piz Escha)
1x Development (Piz Kesch)

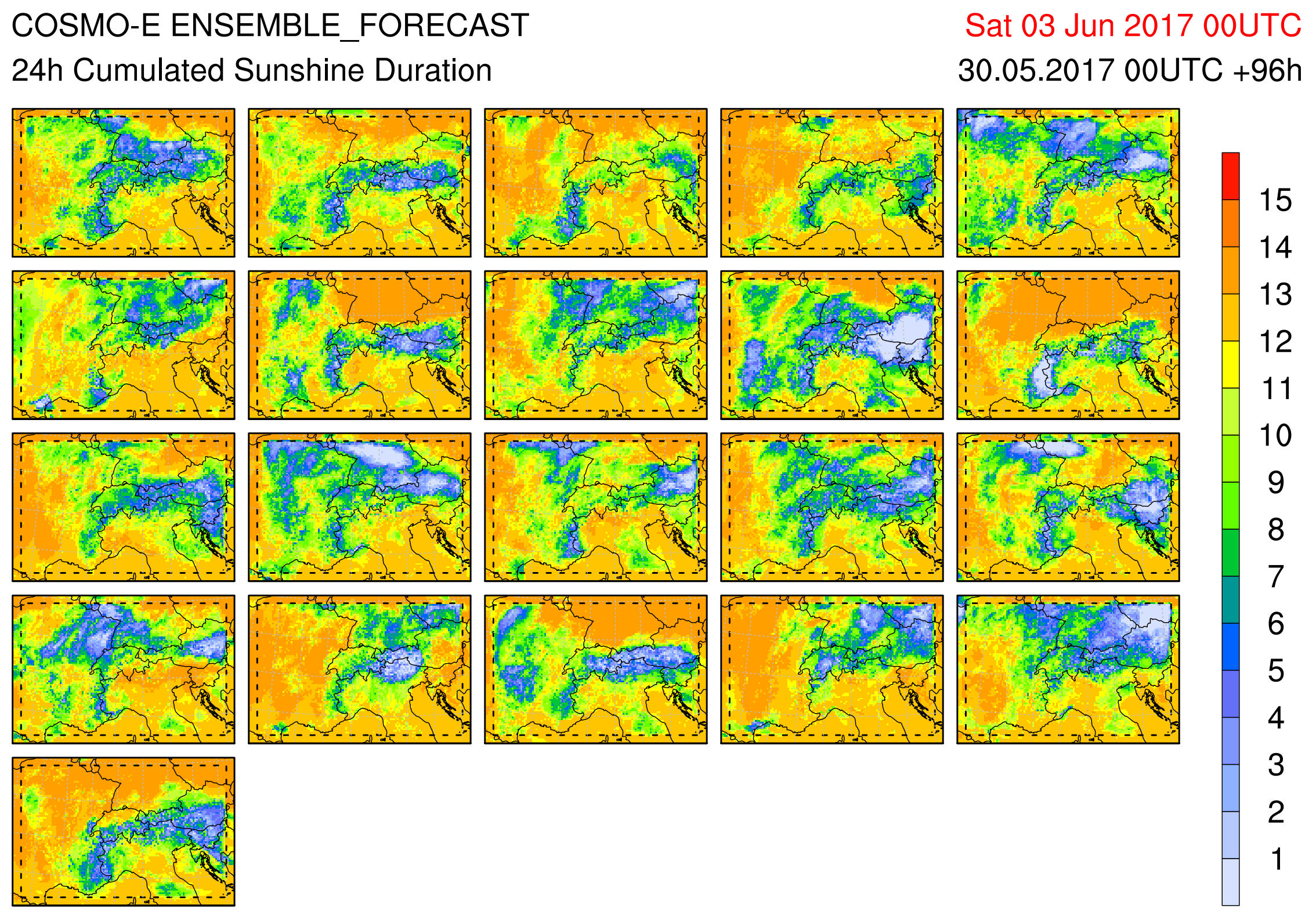
Each rack has
12x Compute nodes with
2x Intel Xeon E5-2690
12 cores @2.6 GHz
256GB RAM
8x NVIDIA Tesla K80 (Stella Duo)
Dual Socket GPUs

5x Post processing nodes

1x MPI Rank: 1x GPU socket, 1x core

Runs

COSMO-E

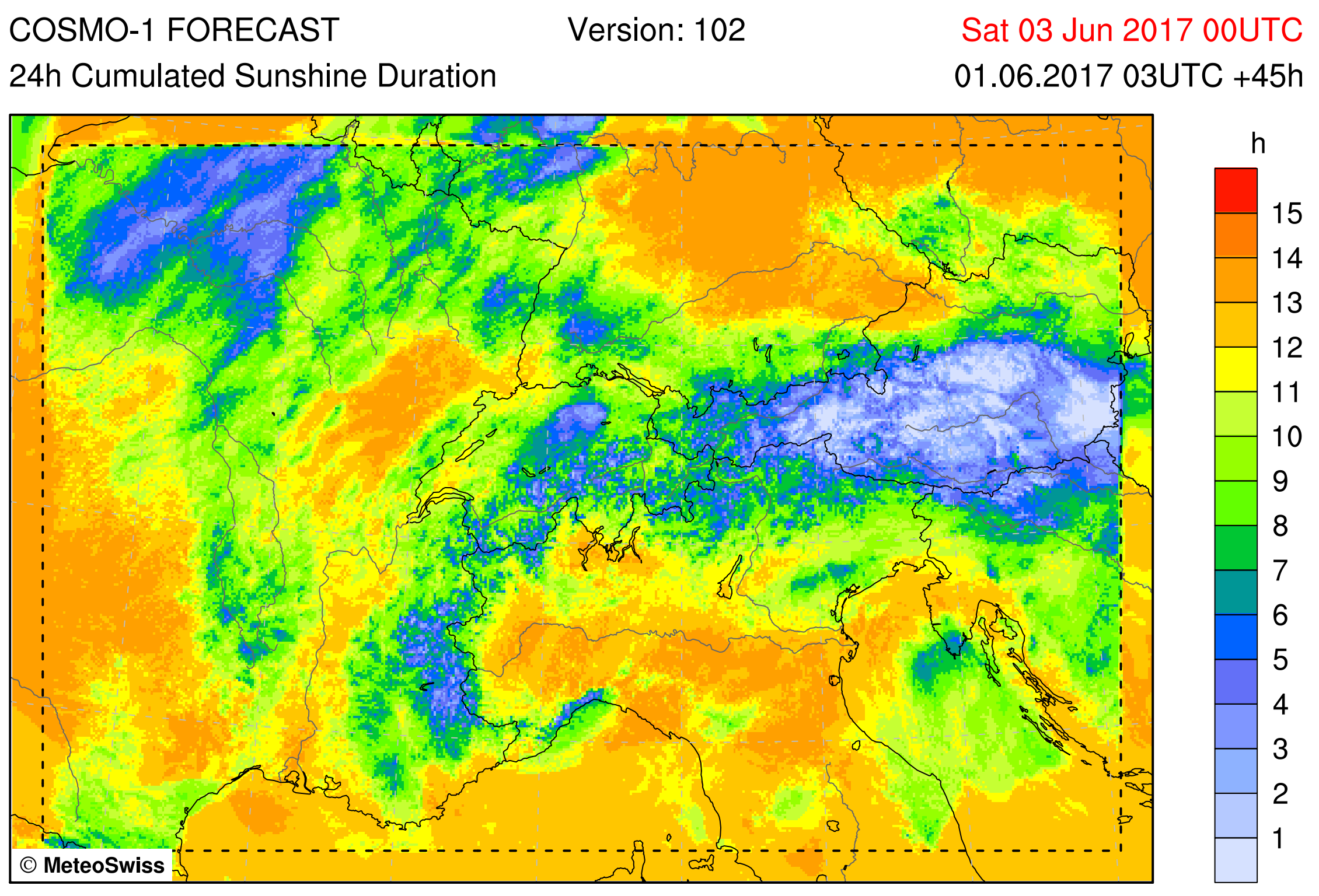


21x10 MPI Ranks: 21 x (8x Compute, 2x I/O)
2.2km resolution (20x members, 1x control)
582x390x60 grid points
120h forecast 2x runs per day (00 and 12 UTC)

Required time to solution 100min for 120h

and

COSMO-1

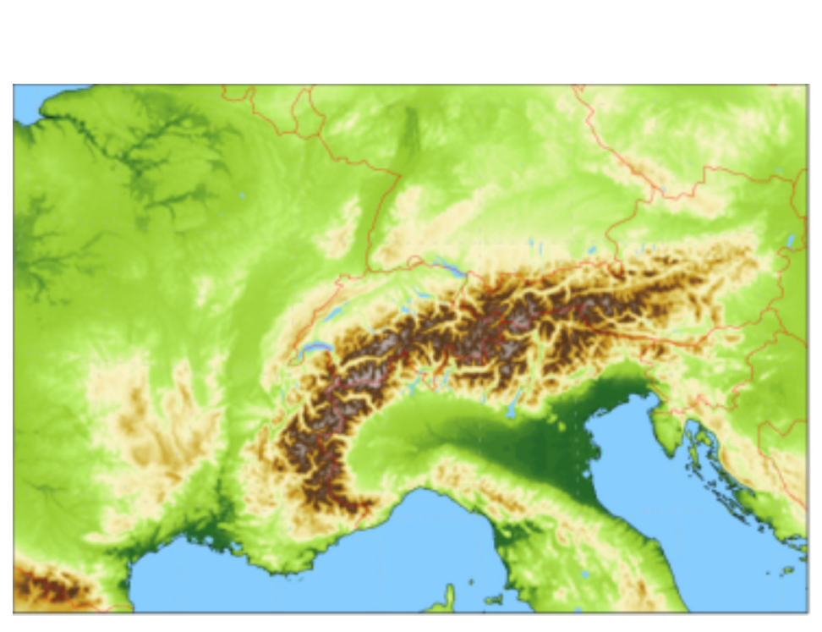


150 MPI Ranks: 144x Compute, 6x I/O
1.1km spatial resolution
1158x774x80 grid points
33h forecast 7x runs per day (00, 06, 09,12, 15, 18, 21 UTC)
45h forecast 1x run per day (03 UTC)

Required time to solution 33min for 33h 45min for 45h

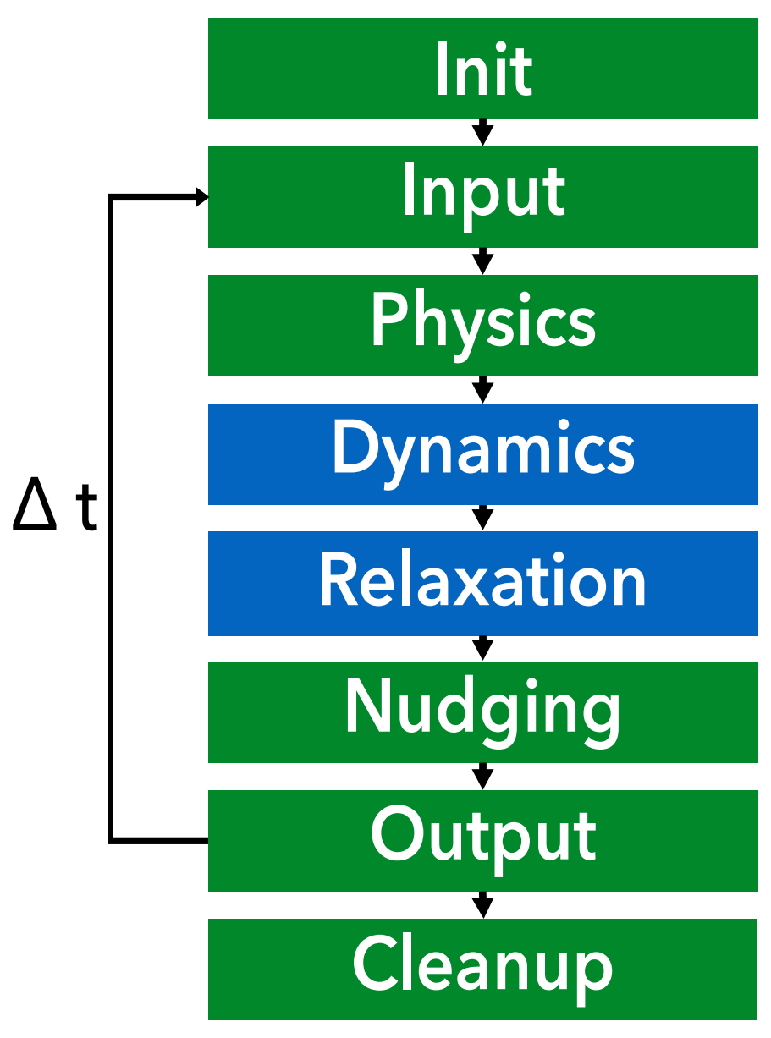
COSMO-E Testcase

1x COSMO-E member
2 hour forecast
Version: 15th May 2017



Baseline

The COSMO Weather Model

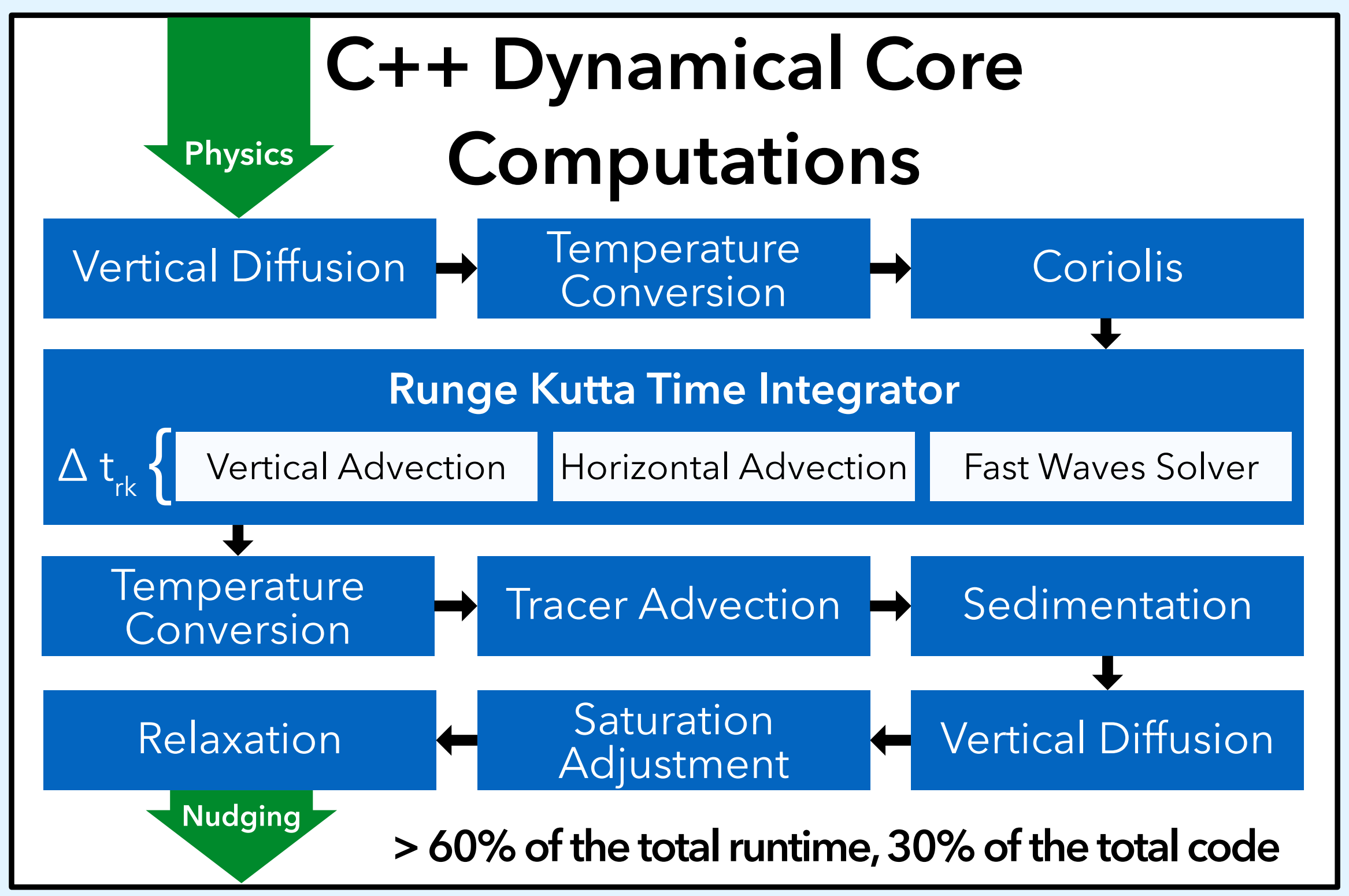


Programming Languages
Fortran: OpenACC
C++: CUDA

The C++ Dycore Rewrite using STELLA

STELLA is a domain specific language directly embedded in C++ geared towards stencil computations. STELLA can generate CUDA, OpenMP and X86 code.

C++ Dynamical Core Computations



> 60% of the total runtime, 30% of the total code

STELLA Stencil Stage Sample

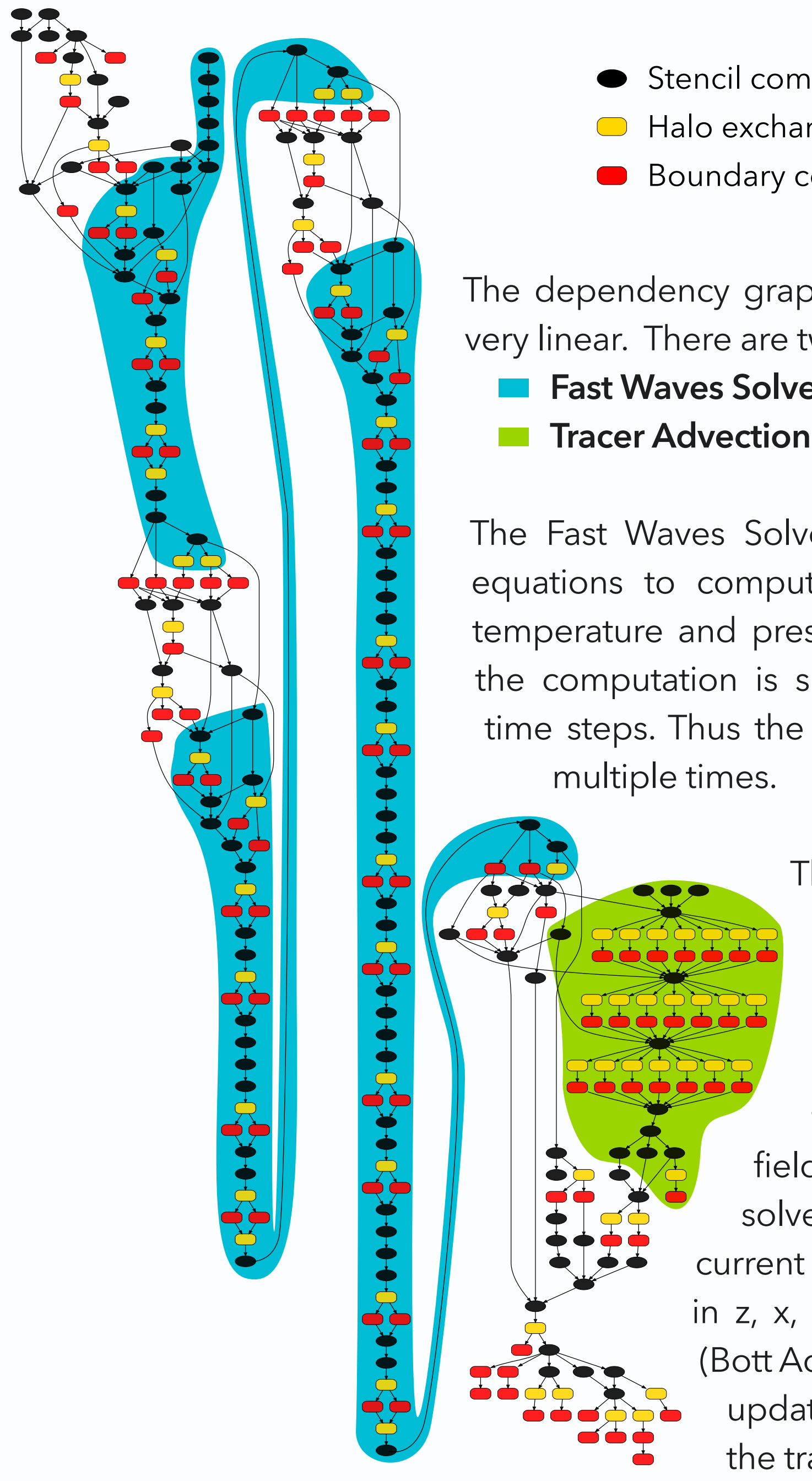
```
template<typename TEnv>
struct LapStage
{
    STENCIL_STAGE(TEnv)
    // Input
    STAGE_PARAMETER(FullDomain, s_in)
    STAGE_PARAMETER(FullDomain, crlato)
    STAGE_PARAMETER(FullDomain, crlatu)
    // Output
    STAGE_PARAMETER(FullDomain, lap)

    __ACC__
    static void Do(Context ctx, FullDomain)
    {
        ctx[lap::Center()] =
            ctx[Call<Laplacian>::With(s_in::Center(),
            /* Cosine ratio in j+1 */ crlato::Center(),
            /* Cosine ratio in j-1 */ crlatu::Center())];
    }
};
```

Computation of the laplacian in the Horizontal Diffusion on the variable s_in.

A stage describes the inner part of an `i,j,k` loop that is computed on a 3D-field. Multiple stages can be combined into loops which then form a stencil.

C++ Dycore Dependency Graph



● Stencil computations
● Halo exchange / Communication
● Boundary condition

The dependency graph of the computations is very linear. There are two areas of interest:
- 🟦 Fast Waves Solver
- 🟩 Tracer Advection

The Fast Waves Solver solves the prognostic equations to compute new values for wind, temperature and pressure. For these variables the computation is split into multiple smaller time steps. Thus the implementation is called multiple times.

The Tracer Advection component is heavy in both computation and communication: It transports tracers after the update of the wind field from the Fast Waves solver. The scheme used in the current version advects the fields in z, x, y, x and then z direction (Bott Advection). Thus requires an update on the boundaries of all the tracers after each direction.

Future Developments

- A model that allows for more computation and communication overlap
- Automatic detection and scheduling of boundary updates
- Improvements in communication to reduce GPU idle time.
- Runtime system to manage memory allocation of temporary fields
- Increase level of abstraction for vertical staggering
- Porting from STELLA to Gridtools

CPU Performance

	Double Precision		
	COSMO Fortran		**COSMO C++ Dycore**
Dynamics/Relaxation	280s	1.32x	207s
Physics	80s		80s
Other	14s		12s
Total	**374s**	1.23x	**309s**

Compute: **8**x Intel Xeon E5-2690 **Sockets** (96x Haswell Cores)
I/O: **4**x Intel Xeon E5-2690 **Cores** (4x Haswell Cores)

1.8x **COSMO C++ Dycore: CPU → GPU**

GPU Performance

	Double Precision		**Single Precision**
Dynamics/Relaxation	98s	1.6x	63s
Physics	38s	1.4x	27s
Other	34s		21s
Total	**169s**	1.5x	**110s**

Compute: **8**x Intel Xeon E5-2690 **Cores** (8x Haswell Cores)
4x NVIDIA Tesla K80 **Cards** (8x Kepler Sockets)
I/O: **2**x Intel Xeon E5-2690 **Cores** (2x Haswell Cores)

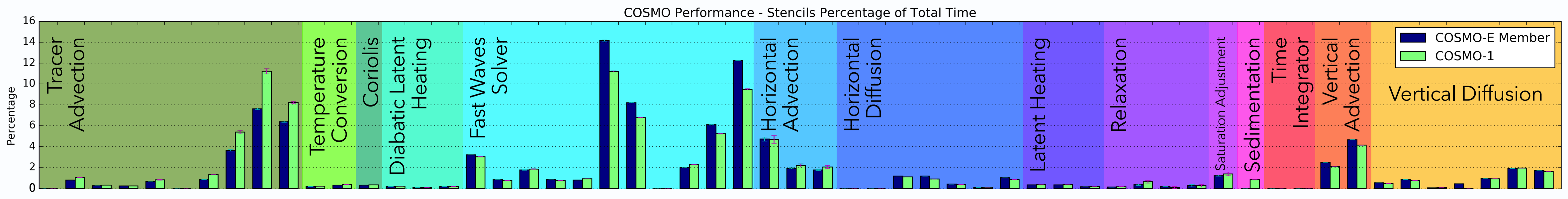
Total Speedup 3.4x

CPU Double Precision (Fortran) ⟶ GPU Single Precision (OpenACC Fortran & C++ Dycore)

STELLA Optimization Levels

GPU Single Precision	**COSMO-E test case**		**COSMO-1 test case**	
No Optimization	93.7s		295.1s	
Merge Stages	84.9s		276.4s	
+ Software Managed Caching	67.2s		238.4s	
+ Parallelize Vertical Levels	65.7s		231.8s	1.29x
+ Texture Caches	64.5s	1.48x	**227.7s**	
+ Blocked Communication	**63.3s**		241.5s	

Distribution of Stencil Computations



Contact

Pascal Spörri
MeteoSchweiz
Operation Center 1
8058 Zürich Flughafen
Switzerland

pascal.spoerri@env.ethz.ch

References

Fuhrer, O., Osuna, C., Lapillonne, X., Gysi, T., Cumming, B., Bianco, M., ... & Schulthess, T. C. (2014). Towards a performance portable, architecture agnostic implementation strategy for weather and climate models. Supercomputing frontiers and innovations, 1(1), 45-62.

Gysi, T., Osuna, C., Fuhrer, O., Bianco, M., & Schulthess, T. C. (2015, November). STELLA: A domain-specific tool for structured grid methods in weather and climate models. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (p. 41). ACM.

Bott, A. (1989). A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. Monthly Weather Review, 117(5), 1006-1016.

Doms, G., & Baldauf, M. (2015). A Description of the Nonhydrostatic Regional COSMO-Model. Deutscher Wetterdienst.

Code Versions and Setup

COSMO	Version 5.0_2017.5 (Basil)	**CPU Setup**	CCE 8.4.4 (PrgEnv-Cray 15.10)
STELLA	Version 1.04.16 (Deneb)		GCC 5.3.0, MVAPICH 2.1
COSMO-1 Case	6h MeteoSwiss reference test case, updated 15. Mai 2017	**GPU Setup**	CCE 8.4.4 (PrgEnv-Cray 15.10, CUDA 7.0)
COSMO-E Case	2h MeteoSwiss reference test case, updated 15. Mai 2017		GCC 4.9.3, CUDA 7.0, MVAPICH 2.1 GDR