Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI
Bundesamt für Meteorologie und Klimatologie MeteoSchweiz

Next Generation ICON NWP forecasting system on GPUs at MeteoSwiss

Xavier Lapillonne

ECMWF – HPC Workshop – 24.09.2021

Roadmap NWP Systems

ICON

ICON-NEST

IFS

ICON-2E

ICON-1E

CONSORTIUM FOR SMALL SCALE MODELING COSMO

IFS

COSMO-2E

COSMO-1E

ICON-Warn

ICON-22

2019 2020 2021 2022 2023 2024 2025 2026 2027

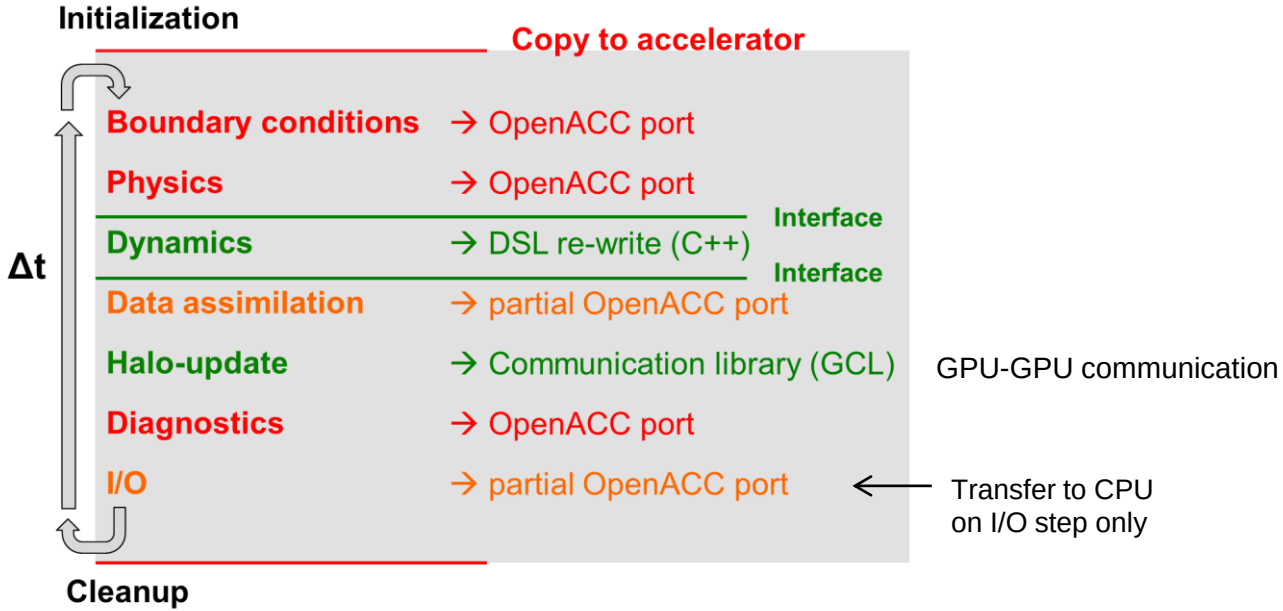
Use of software based infrastructure

Tsa/Arolla

1 Cray CS-STORM GPU-System

COSMO model on GPU

- Local area numerical weather prediction and climate model (cosmo-model.org), 350 KLOC F90 + MPI
- Full GPU port strategy : avoid GPU-CPU data transfer
- Approach: OpenACC compiler directives + Domain Specific Language (DSL) re-write
- 4-5x faster on GPU (socket to socket comparison)



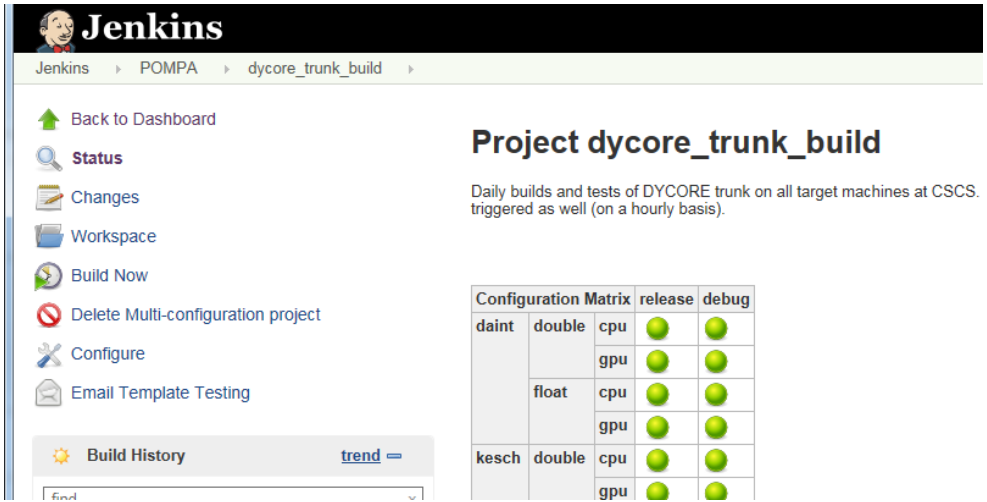
Software development

- GPU support increase complexity of the software
- Need to test many configurations on CPU, GPU, with/withou DSL, single precision, …

➡ Improve software development process : code review, automatic on demand and nightly build and tests (Jenkins, buildbot)

➡ serialized data for «component» testing

➡ Package management with spack



Lessons learned from COSMO port to GPU

- Stable operation at MeteoSwiss since 2016, 2 generations Cray Hybrid systems

OpenACC:

\+ Incremental insertion in existing code, good acceptance by domain scientist

\- Some compiler bugs/issues, legacy code: no unittest, not always performance portable, only nvidia GPU (at this point)

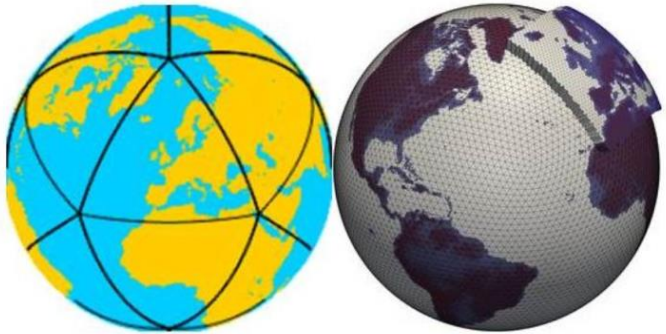
DSL dycore:

\+ Separation of concerns, performance portable : tested on CPU, Intel MIC, Nvidia GPU, AMD GPU , easier to test and maintain (unittest)

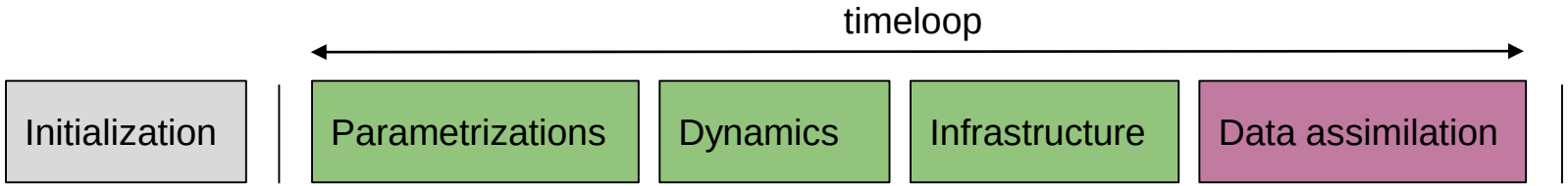
\- Low acceptance, less user friendly, requires new know how, limited to DSL supported features

ICON port to GPU

- ICON: Non-hydrostatic global and regional unified climate and numerical weather prediction model.
- ICON partners: DWD, MPI-M, DKRZ, KIT – ICON dev. Partners: COSMO, C2SM …
- Initial GPU port: OpenACC only, accepted by all partners, step by step integration in official version
- DSL implementation of components - research projects : ESCAPE, ESiWACE, EXCLAIM



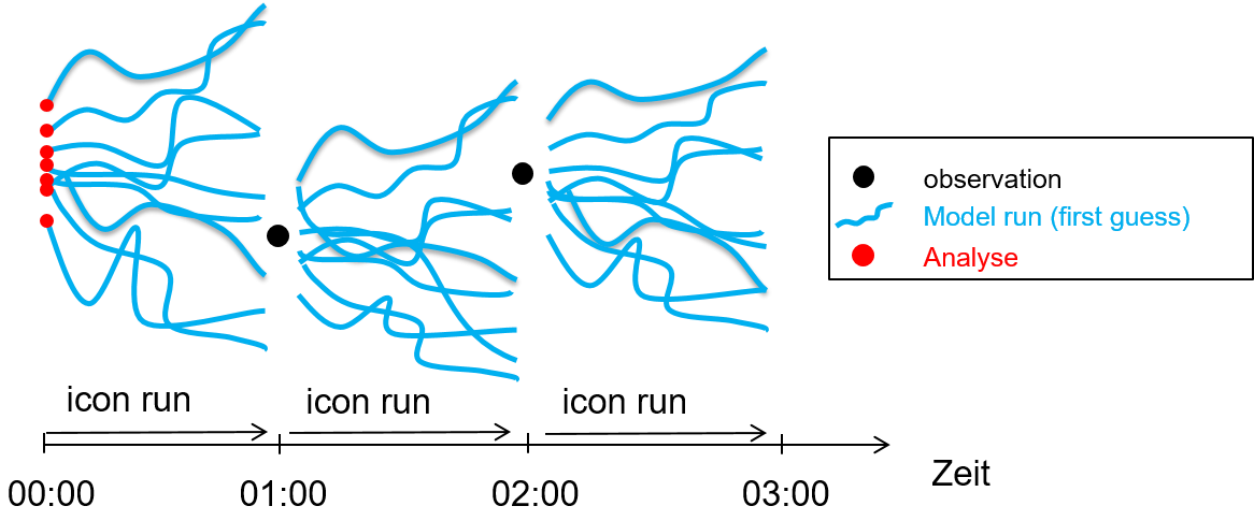
Porting strategy ICON: OpenACC



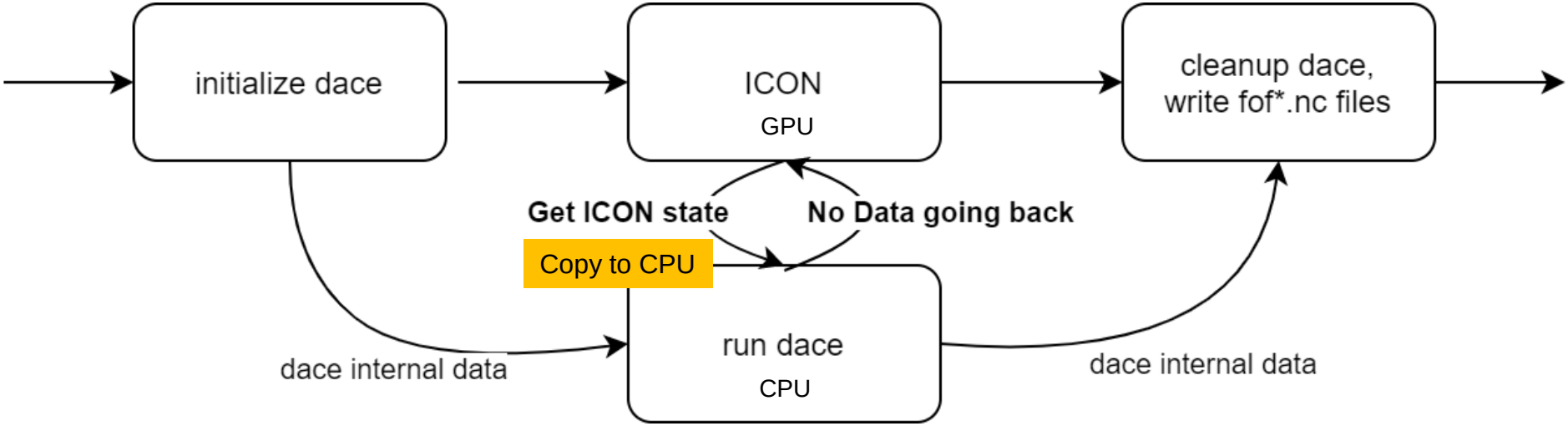
- Avoid GPU-CPU transfer: all components of the time loop need to be ported to GPU
  - Exception: Data assimilation runs on CPU
- Design : Main parallelization along horizontal blocking (nproma - not at the block level like OpenMP for CPU)
  - Use large nproma when running on GPU (ideally 1 single block per GPU)
  - Compatible with COSMO parameterizations already ported to GPU
- Testing:
  - Individual components with Serialbox and PCAST (Nvidia compiler tool) during development
  - Comparing model output with tolerance (CPU/GPU not bit-identical)
- Investigation for AMD GPU just started : OpenMP, translation tools, gcc, …

ICON with Data Assimilation on GPU

- Kilometer-Scale Ensemble Data Assimilation (KENDA). Calculations in ensemble space spanned by the ensemble members, using observations to compute analysis
- Assimilation component takes the model and compares them with observations (DACE) – wirte feedback files (fof)



Data Flow, and GPU strategy



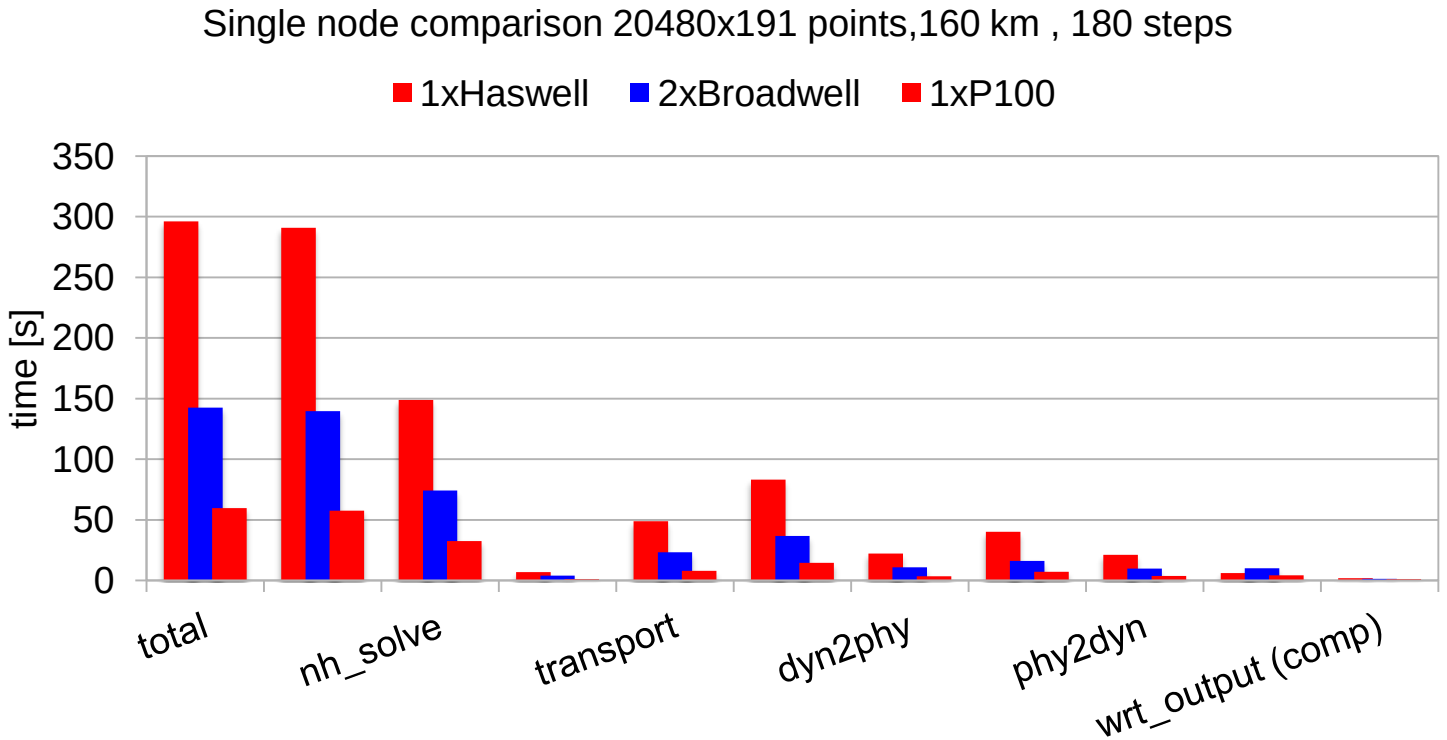
- DACE code is not ported to GPU. The DACE code is kept on the CPU. Data is copied from GPU to CPU when needed.

Porting status ICON

- Ready for some global climate application (QUBICC configuration – MPI-M)
- NWP configuration : on going, should be operational in 2023

Status NWP	ported	optimized
nh_solve		
nh_hdiff		
transport		
convection		
LHN		
radiation		
radheat		
microphysics		
Cloud cover		
turbulence		
surface		
sso		
two-way nesting		

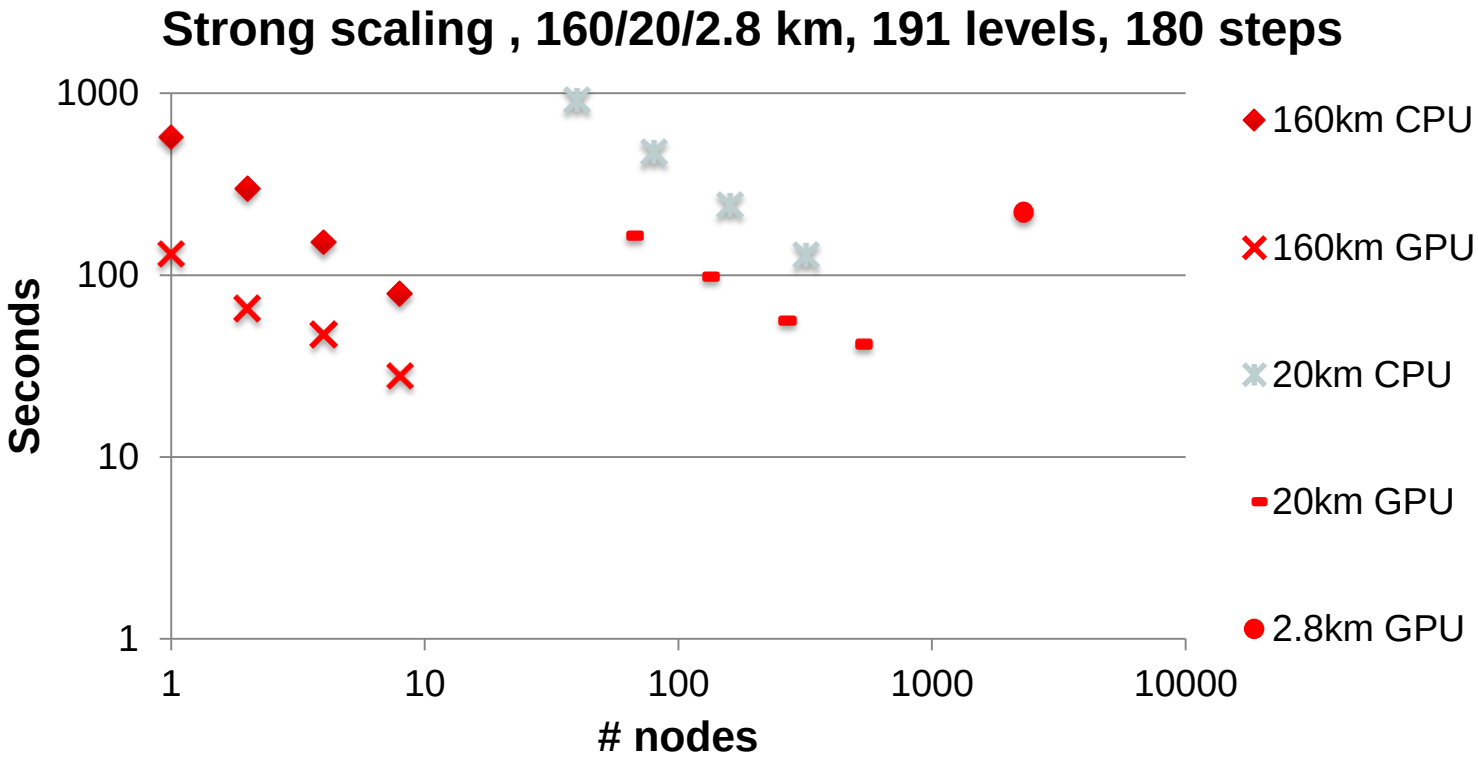
Performance results: single node (QUBICC conf.)



Single socket speedup, 1 Haswell CPU vs1 P100 GPU: 4.9x

Source :W. Sawyer CSCS

Performance : strong scaling (QUBICC conf.)



CPU: 1xHaswell, 12 ranks/node, OMP=2 (CCE)
GPU: 1xP100 (daint-gpu), 2 ranks/node (PGI)

DSL for ICON

Motivation

$$\nabla_{\underline{n}}\psi(e) = \frac{\psi(c_1(e)) - \psi(c_0(e))}{\hat{l}}$$

```
#ifdef _OMP
!$OMP ....
#else
!$ACC ....
#endif
DO jb = i_startblk, i_endblk
 CALL get_indices_e(ptr_patch, ...)
 #ifdef __LOOP_EXCHANGE
 DO je = i_startidx, i_endidx
   DO jk = slev, elev
 #else
   DO jk = slev, elev
     DO je = i_startidx, i_endidx
 #endif
     grad_norm_psi_e(je,jk,jb) =  &
       ( psi_c(iidx(je,jb,2),jk,iblk(je,jb,2)) -
         psi_c(iidx(je,jb,1),jk,iblk(je,jb,1)) )
       / ptr_patch%edges%lhat(je,jb)
   ENDDO
 END DO
END DO
#ifdef _OMP
!$OMP ...
#else
!$ACC ...
#endif
```

Motivation

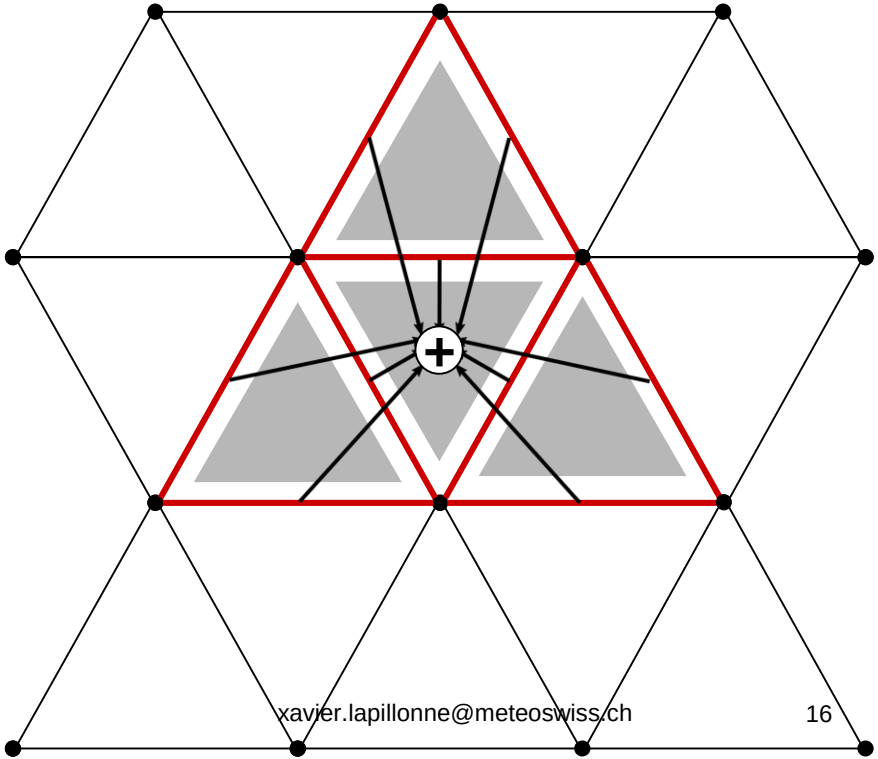
$$\nabla_{\underline{n}}\psi(e) = \frac{\psi(c_1(e)) - \psi(c_0(e))}{\hat{l}}$$

```
grad_norm_psi_e =
      sum_over(psi_c,
               Edge > Cell,
               [1/lhat, -1/lhat] )
```

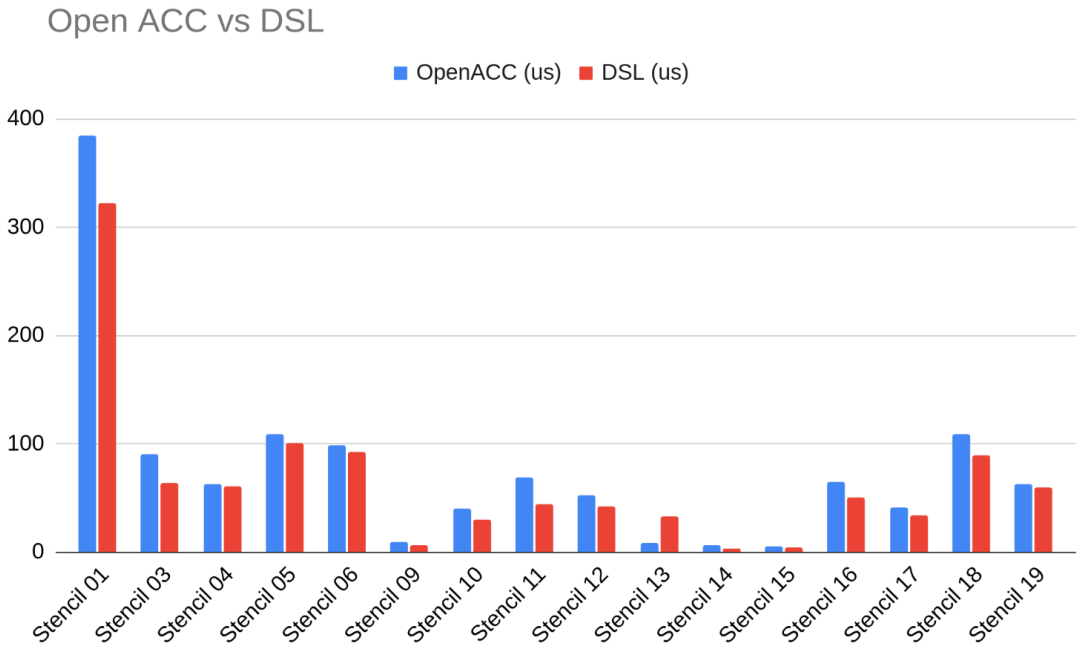
```
#ifdef _OMP
!$OMP ....
#else
!$ACC ....
#endif
DO jb = i_startblk, i_endblk
 CALL get_indices_e(ptr_patch, ...)
 #ifdef __LOOP_EXCHANGE
 DO je = i_startidx, i_endidx
   DO jk = slev, elev
 #else
   DO jk = slev, elev
     DO je = i_startidx, i_endidx
 #endif
     grad_norm_psi_e(je,jk,jb) =  &
       ( psi_c(iidx(je,jb,2),jk,iblk(je,jb,2)) -
         psi_c(iidx(je,jb,1),jk,iblk(je,jb,1)) )
       / ptr_patch%edges%lhat(je,jb)
   ENDDO
 END DO
END DO
#ifdef _OMP
!$OMP ...
#else
!$ACC ...
#endif
```

Python DSL (Dusk/Dawn) notation
example : Neighbor Chains

```
@stencil
def intp(fc: Field[Cell],
         fe: Field[Edge],
         w: Field[Cell > Edge > Cell > Edge]):
    with levels_downward:
        fc = sum_over(Cell > Edge > Cell > Edge,
                      w*fe)
```



Performance and dycore (DSL) prototype



DSL : Dusk/ Dawn

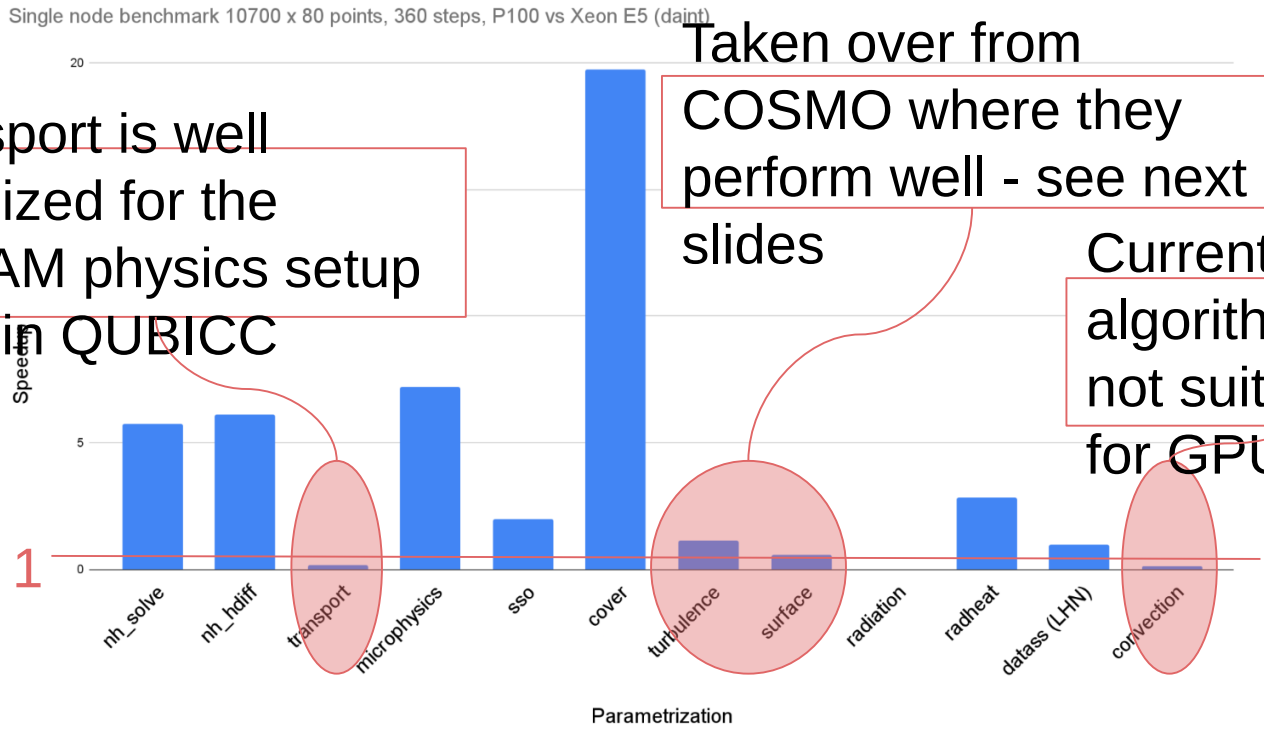
- DSL about 10 to 20% faster then OpenACC - not fully optimized

Conclusions

- COSMO model has been run on GPU since 2016 at MeteoSwiss and will soon be replaced by ICON
- The ICON model is being ported to GPU for NWP applications, first port using OpenACC compiler directives
- It is critical to well understand the data flow, most components of the time loop need to be ported to GPU
- Ongoing effort to develop a python DSL system for ICON – ESCAPE, EXCLAIM
- Exchange and discussion between software engineers and domain scientist is key to success

Additional slides

Status of OpenACC performance



Transport is well optimized for the ECHAM physics setup used in QUBICC

Taken over from COSMO where they perform well - see next slides

Current algorithm likely not suited well for GPU

C2SM / EXCLAIM project

EXascale Computing platform for cLoud-resolving weAther and clImate Models

Goals:

- develop an extreme scale computing and data platform for cloud resolving weather and climate modeling – prepare for exascale system
- redesign the codes in the framework of python base domain-specific languages
- exploit efficiency gains at both the computational and algorithmic levels
- develop an analysis and storage framework to deal with the exabytes of data generated
- design concrete applications addressing scale-interactions in the ocean and atmosphere
- Performance target : 1 simulated year / day @ 1 km global

Large project : 1. Senior scientist, 2 post-docs, 6 Software eng. + in in kind contributions from ETHZ, CSCS, EMPA, SDSC and MeteoSwiss

Multi-core vs. GPU-accelerated hybrid

	Piz Dora (old code)	**Piz Kesch (new code)**	Factor
Sockets	~26 CPUs	~7 GPUs	**3.7 x**
Energy	10 kWh	2.1 kWh	**4.8 x**

Performance Results

- Test Setup as close as we can get right now to the operational setup:
  - Ca. 1.2M Grid columns (Area over Switzerland & Germany)
  - Simulation time of one hour
  - Run on Piz Daint 32 GPU nodes

	GPU run
ICON total [s]	**1214**
DACE total [s]	**6.95**
DACE total [%]	**0.57**

KENDA Assimilationszyklus