Global climate simulations at 2.8 km on GPU with the ICON model

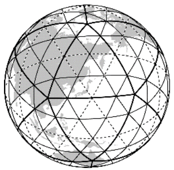
X. Lapillonne, M. Girogetta, W. Sawyer, R. Dietlicher, V. Clement, P. Marti, C. Osuna, S. Ferrachat, L. Kornblueh, M. Esch, R. Schnur, S. Rast, G. Hime, J.F. Engels, G. Zangl, D. Reinert

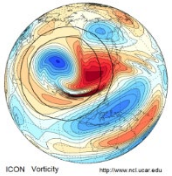
EGU 2020

ICON ENIAC (Enabling ICON on GPU)

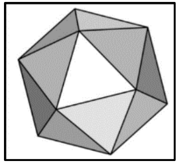
Focus on global climate modeling

- The porting work on the ICON model is part of the ENIAC project
- Adapt the global weather and climate model ICON to run on GPU and many-core architecture
  - Base line GPU port with OpenACC compiler directives
- Prepare the ICON model for actual use-cases in global climate on emerging HPC systems
- Achieving a high degree of performance portability by:
  - using source-to-source translation tool : CLAW
  - using alternative approaches such as the GridTools domain-specific language

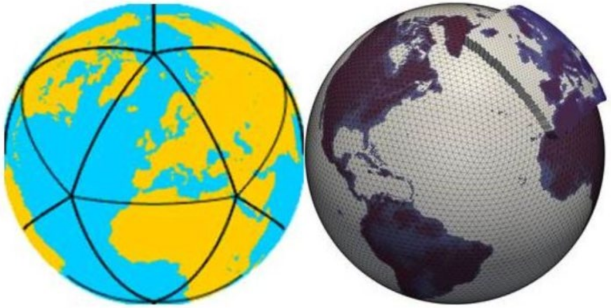




ICOsahedral Nonhydrostatic model



- Non-hydrostatic global unified climate and numerical weather prediction model developed by DWD (for NWP) and MPI-Hamburg (for climate)
- Next generation model replacing COSMO, GM,ECHAM,
- Icosahedral Grid: quasi uniform grid resolution with optional regional refinement
- Large community code Fortran + MPI + OpenMP base code, over 1 Million LOC
- Currently used for operational weather forecast at DWD, and in many Universities for climate modeling
- Main future climate model for C2SM members: ETHZ, MeteoSwiss, Empa

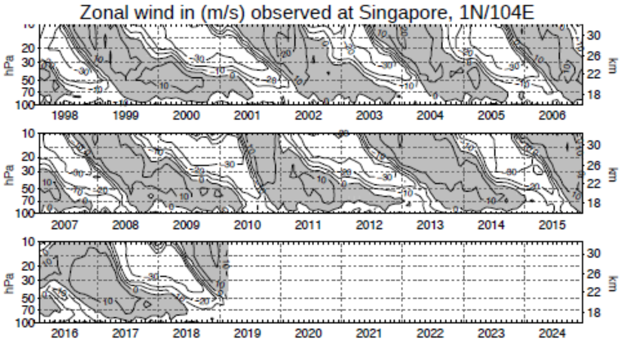


Target application at 2.8 km : QUBICC (MPI-M)

- "Quasi-Biennial Oscillation in a Changing Climate (QUBICC)“
- The QBO winds as observed for ~60 years are fairly regular, except for the breakdown in early 2016. How is QBO affected in a changing climate
- PRACE proposal accepted to run global simulations at 2.8 km horizontal resolutions on Piz Daint GPU system. First test runs started.

Modell configuration

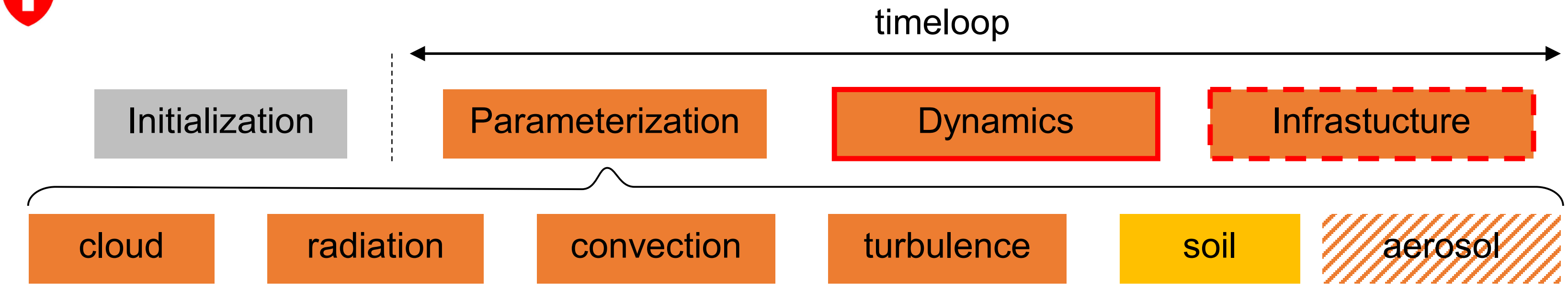
- Dynamics/transport on a 2.8 km grid (R2B10) with ca. 200 layers up to ca. 80 km.
- Radiation, Vertical diffusion implicitly coupled to JSBACH-lite"Graupel" cloud microphysics & saturation adjustment



FU-Berlin, https://www.geo.fu-berlin.de/met/ag/strat/produkte/qbo/qbo_wind.jpg

Credit: Marco Giorgetta MPI-M

Porting Strategy : full GPU port



- First base version OpenACC:
  - Dynamics + Climate physics + infrastucture: OpenACC
  - JSBACH (soil model) : claw-dsl
  - Dynamics and part of the infrastructure already implemented by CSCS (W. Sawyer) prior to ENIAC
- Required code adaptation and changes
- Only supported compiler for GPU : PGI (requirement for OpenACC 2.6)
- Implement new tolerance base testing, integrated in regular automatic ICON testing infrastructure

Experience porting ICON with OpenACC

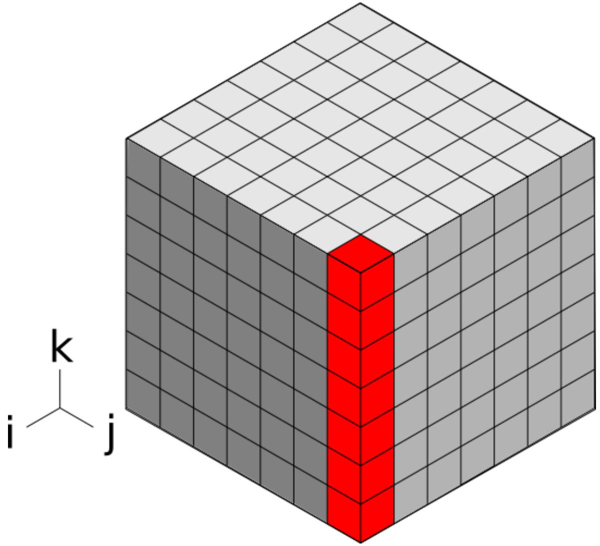
- Difficult to port a large and fast evolving model : ~800 Fortran module files, ~1mio LOC
- Many compiler issues (deep copy, atomics, non supported fortran 2003 feature …): OpenACC+Fortran is not a widely used technology.
- Difficult pattern to port, e.g. large derived type, Object oriented Fortran
- Limited compiler support for newer standard version >= 2.6 (only PGI)
- Validation of GPU vs. CPU required to implement testing infrastucture

CLAW-DSL for JSBACH (soil) model

- CLAW: Fortran Source-to-source translator based on the OMNI Compiler Project
- CLAW-DSL : targets physical parameterization : single column computation
- JSBACH : Fortran 2003/2008 code style - ~25k LOC
- Elementals function are a special case of single column abstraction with no vertical loop : use CLAW-DSL
- CLAW port : ~140 EXPAND kernels automatically generated, ~90 ELEMENTAL (SCA) kernels automatically generated, ~100 files parsed by the CLAW Compiler, ~20 files with deep transformation
- Port of individual JSBACH tasks completed, integration in ICON ongoing

CLAW DSL: Clement V, Proceedings of the Platform for Advanced Scientific Computing Conference. ACM, 2018

CLAW Single Column Abstraction (SCA)



Targets physical parameterization

- No notion of independent horizontal dimension
  - No DO statements over horizontal
  - Arrays are demoted to get rid of the horizontal

Separation of concerns

- Domain scientists focus on their problem (1 column, 1 box)
- CLAW Compiler produce code for each target architecture and directive languages

Code example (original + CLAW directive)

CLAW directives

```
PURE ELEMENTAL SUBROUTINE calc_radiation_surface_net(swvis_down, swnir_down, &
  alb_vis, alb_nir, lw_down, t, rad_net, swvis_net, swnir_net, sw_net, lw_net)

  USE mo_phy_schemes, ONLY: lwnet_from_lwdown

  !$claw model-data
  REAL(wp), INTENT(in) :: swvis_down, swnir_down, alb_vis,  alb_nir, lw_down, t
  REAL(wp), INTENT(out) :: rad_net
  REAL(wp), INTENT(out), OPTIONAL :: swvis_net, swnir_net, sw_net, lw_net
  !$claw end model-data
  !$claw sca

  REAL(wp) :: zswvis_net, zswnir_net, zsw_net, zlw_net

  ! Compute net SW radiation from downward SW and albedo
  zswvis_net = swvis_down * (1._wp - alb_vis)
...
```

Code example (transformed)

clawfc -D__ICON__ --model-config=icon_jsbach.toml --target=gpu --directive=acc -o code_transformed.f90 original.f90

```
SUBROUTINE calc_radiation_surface_net ( swvis_down , swnir_down , &
 alb_vis,  &
 alb_nir , lw_down , t , rad_net , swvis_net , swnir_net , sw_net , lw_net)
 USE mo_phy_schemes , ONLY: lwnet_from_lwdown
 INTEGER , INTENT(IN) :: kproma

 REAL ( KIND= wp ) , INTENT(IN) :: swvis_down ( : )
 REAL ( KIND= wp ) , INTENT(IN) :: swnir_down ( : )
 REAL ( KIND= wp ) , INTENT(IN) :: alb_vis ( : )
 REAL ( KIND= wp ) , INTENT(IN) :: alb_nir ( : )
 REAL ( KIND= wp ) , INTENT(IN) :: lw_down ( : )
 REAL ( KIND= wp ) , INTENT(IN) :: t ( : )
 REAL ( KIND= wp ) , INTENT(OUT) :: rad_net ( : )
…
```

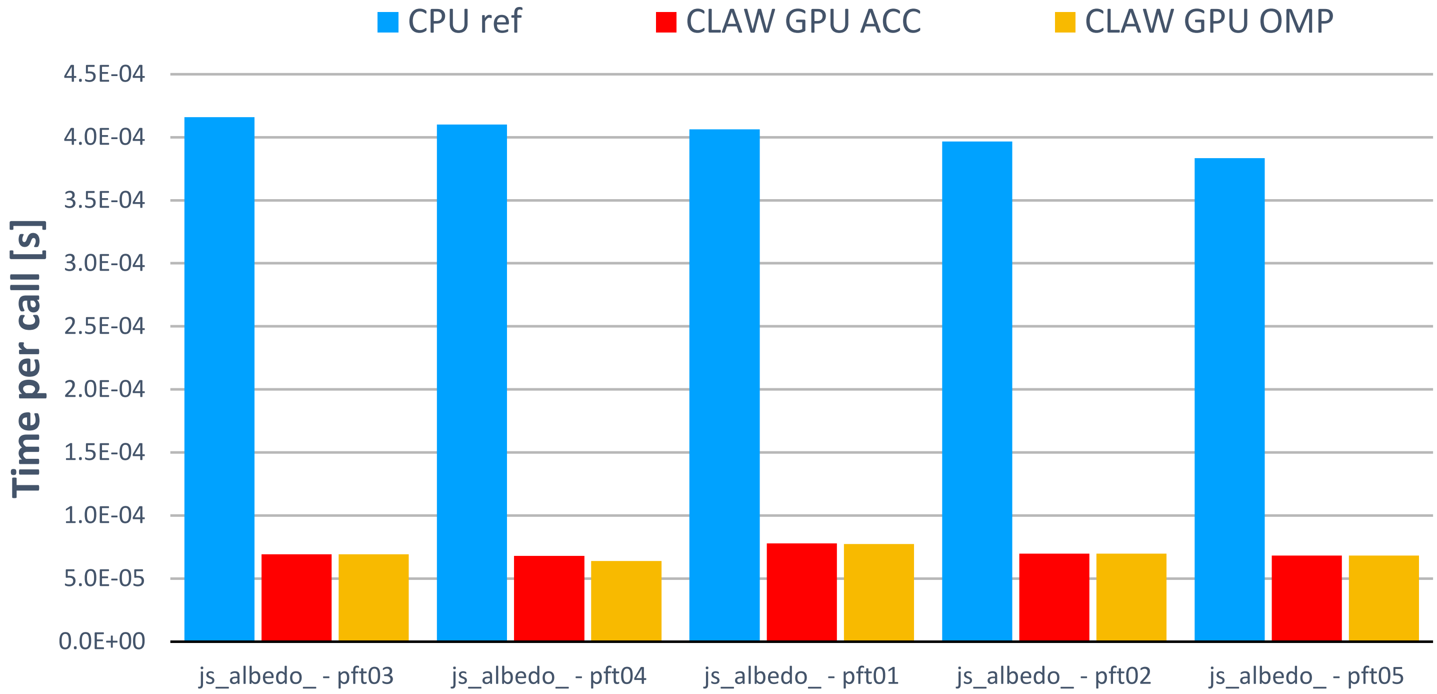
```
…
!$acc data &
!$acc present(swvis_down,swnir_down,alb_vis,alb_nir,&
!$acc ,  lw_down,t,rad_net &
!$acc ,swvis_net,swnir_net,sw_net,lw_net)
!$acc parallel
!$acc loop gang vector
  DO horizontal = 1 , kproma , 1
   zswvis_net = swvis_down ( horizontal ) * ( 1._wp - alb_vis …
   zswnir_net = swnir_down ( horizontal ) * ( 1._wp - alb_nir  …
   zsw_net = zswvis_net + zswnir_net
   zlw_net = lwnet_from_lwdown ( lw_down ( horizontal ) , …
   rad_net ( horizontal ) = zsw_net + zlw_net
…
```

Performance Results JSBACH/CLAW

- For GPU claw can generate OpenACC or OpenMP for accelerator directives
- OpenACC results obtained using the PGI compiler, timed in full application
- OpenMP for accelerator : Cray. Note : due to current limitation and issues with the compiler the results are obtained with standalone kernels

Performance results

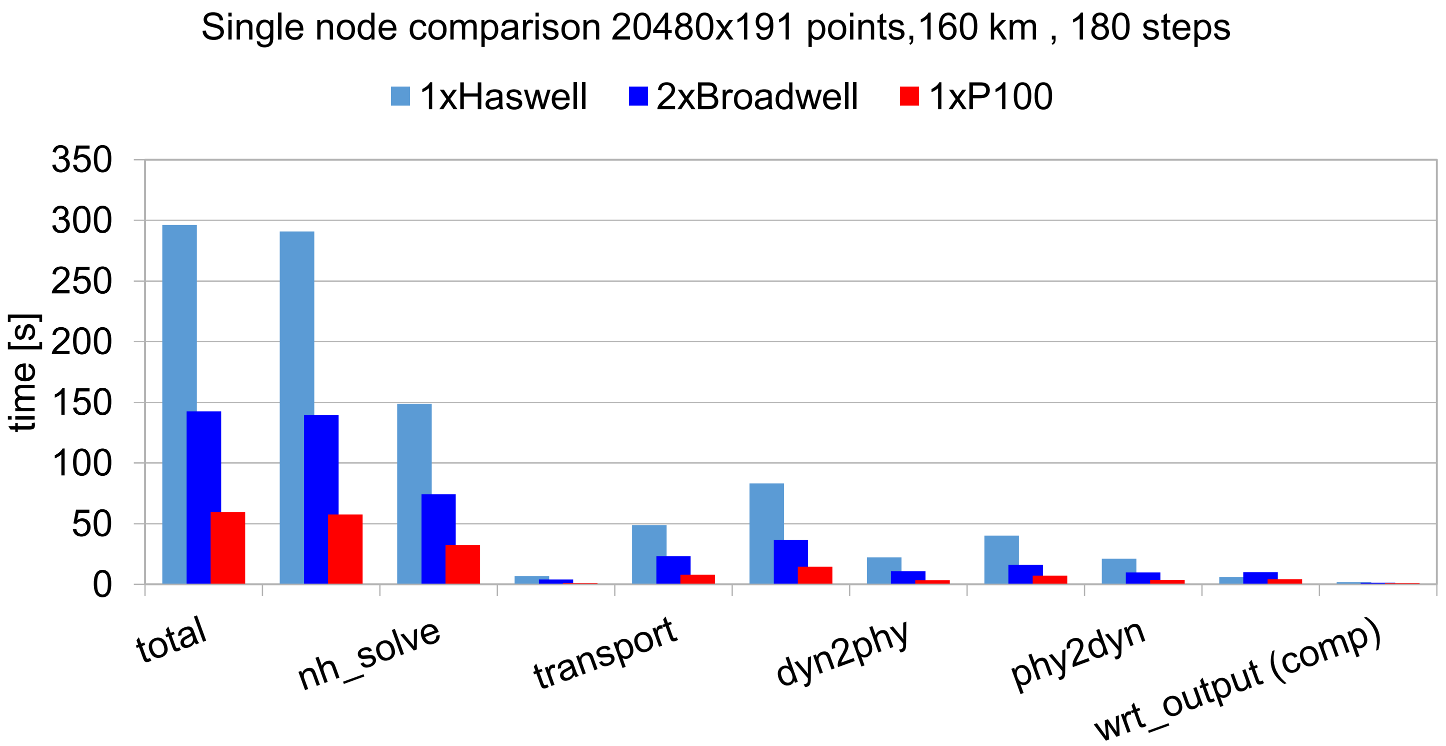
Performance comparison (socket to socket) of 5 JSBACH tasks on Intel Haswell E5-2690v3 and NVIDIA P100. Domain size (horizontal grid points x vertical levels) = 20480 x 47



Speed up ca 3x-8x. CLAW-DSL, JSBACH: ISC'19 Best Poster Award, Clement, Superfri 2019

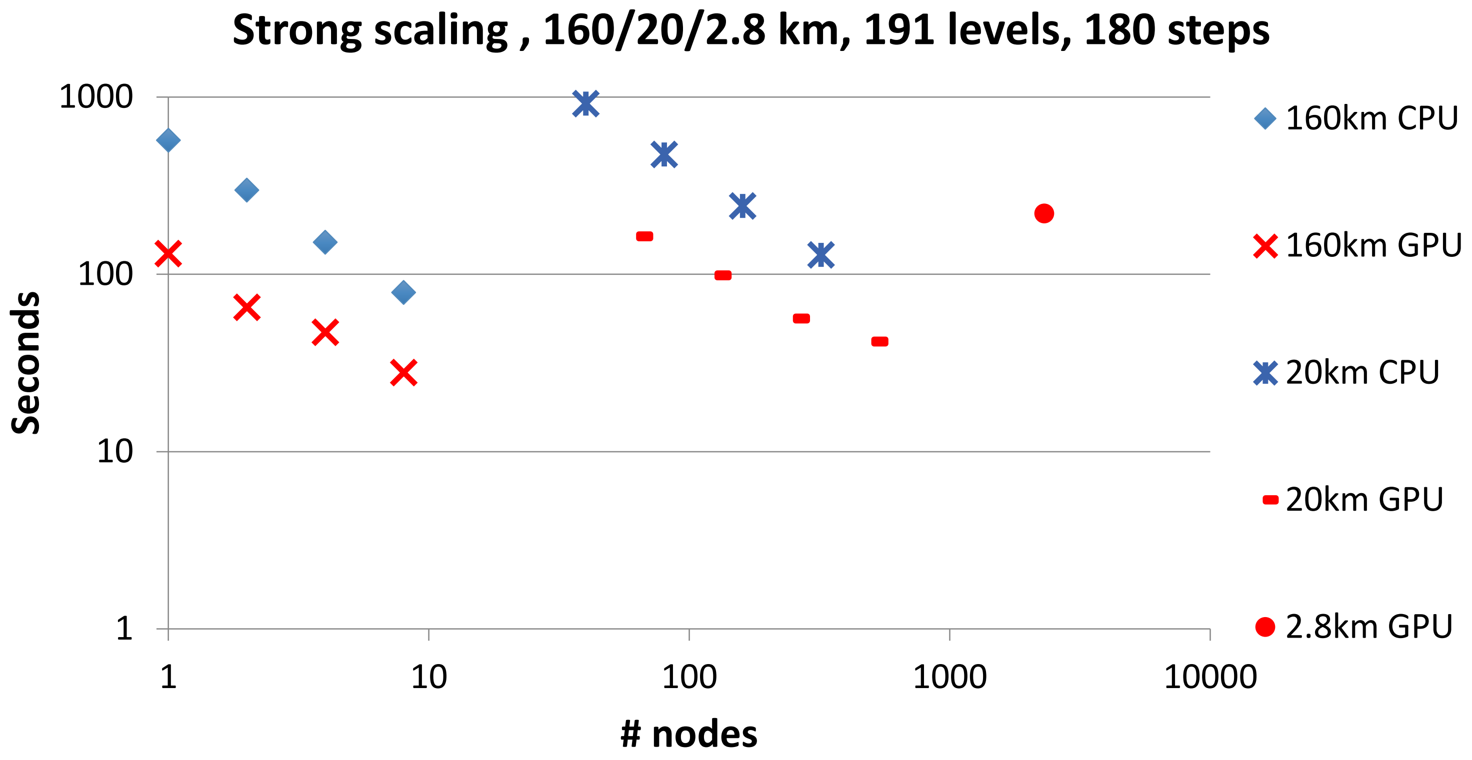
Performance full model

- Performance results have been obtained using the full model with I/O
- Runs have been carried out on the Piz Daint computer at CSCS



Single socket speedup, 1 Haswell CPU vs1 P100 GPU: 4.9x
Source :W. Sawyer CSCS

Performance : strong scaling



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

Preliminary results at 2.8km

- Use of GPU-GPU communication improve performance by 5-10%
- First test simulation at 2.8 km on 2300 P100 GPU nodes on Piz Daint
- Realistic I/O, using 12 I/O nodes
- Simulation Year Per Day : 0.04 SYPD
- Scaled up to 2.8 km setup and 30 months : $Tf = 2.5 \cdot 10^6$ node·h

Main achievements

- Full port of the ICON model to GPU for first climate application (e.g. QUBICC project), including all required infrastructure, e.g. I/O, communication …
- All changes integrated in the latest release candidate, will be available for the entire ICON community
- Strong collobaration with PGI/Nvidia, many bugs reported to the compiler
- Explore abstraction and portability using the CLAW-DSL for soil model
- CLAW compiler : open source project https://github.com/claw-project