Accelerating tracers in FESOM2 on GPU's

Gijs van den Oord, David Guibert, Alessio Sclocco, Erwan Raffin, Ben van Werkhoven, Natalya Rakowski, Nicolay Koldunov, Dmitry Sidorenko

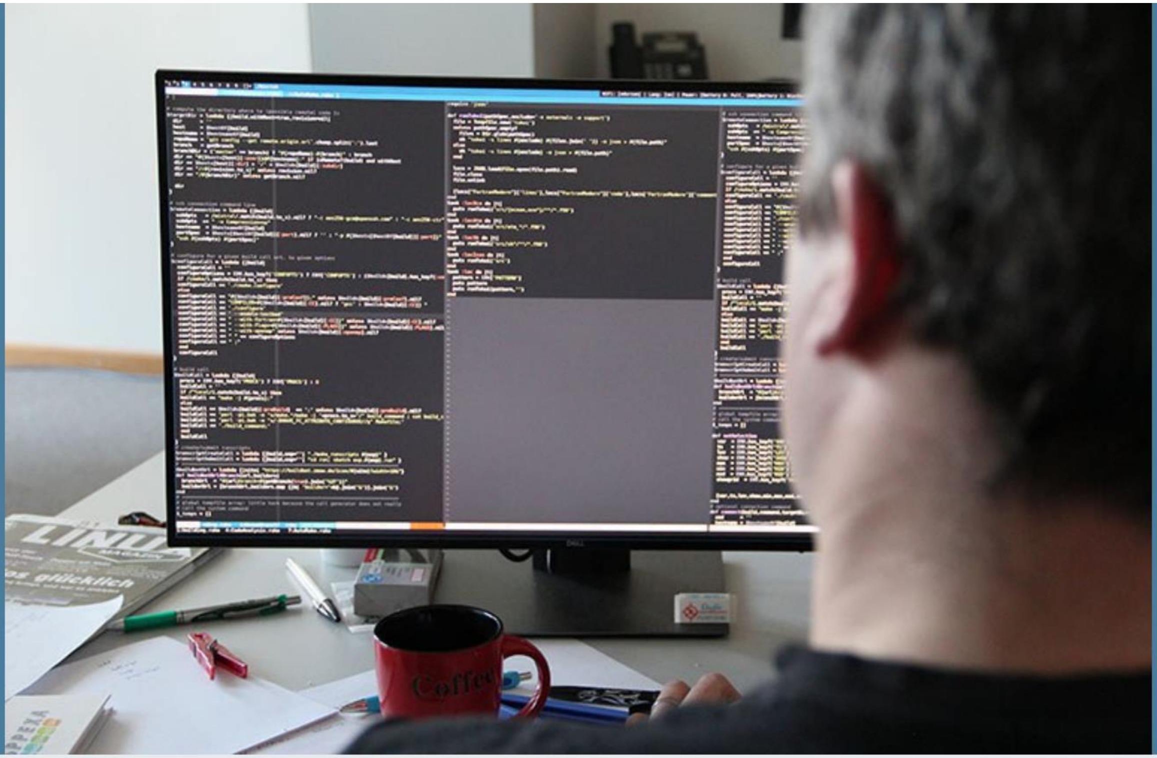
netherlands eScience center

Atos

esiwace2
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Model Refactoring and Porting

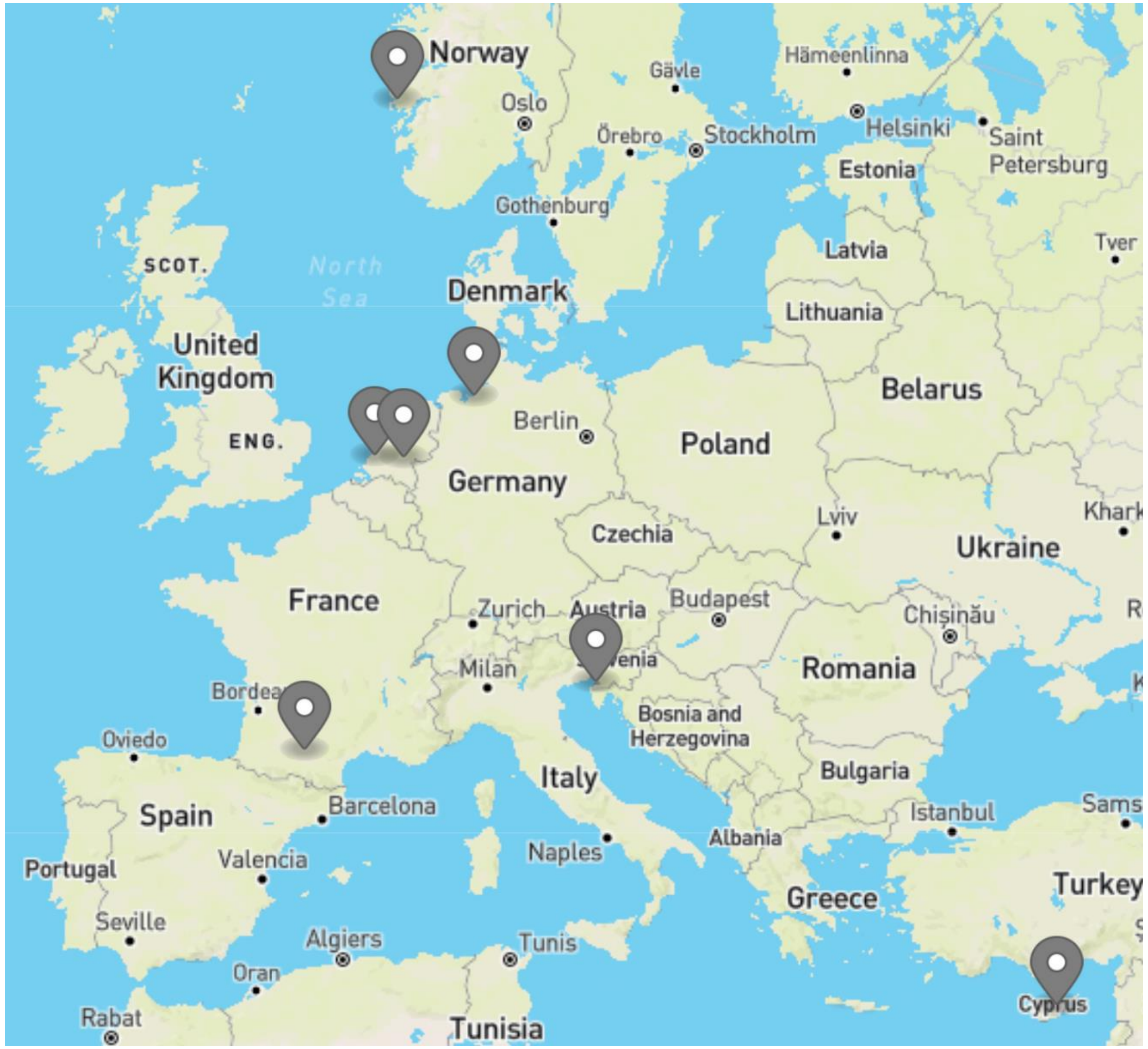
Apply for support with refactoring or porting your weather and climate code to GPUs!



This service supports the exascale preparations of the weather and climate modelling community in Europe. We create short collaboration projects that provide guidance, engineering, and advice for improving model efficiency and for porting models to existing and upcoming computing infrastructures. All groups developing and maintaining weather and climate codes - not only the ESiWACE2 partners - can apply. The projects, funded by ESiWACE2, will be granted in-kind contributions by the Netherlands eScience Center and/or Atos-Bull.

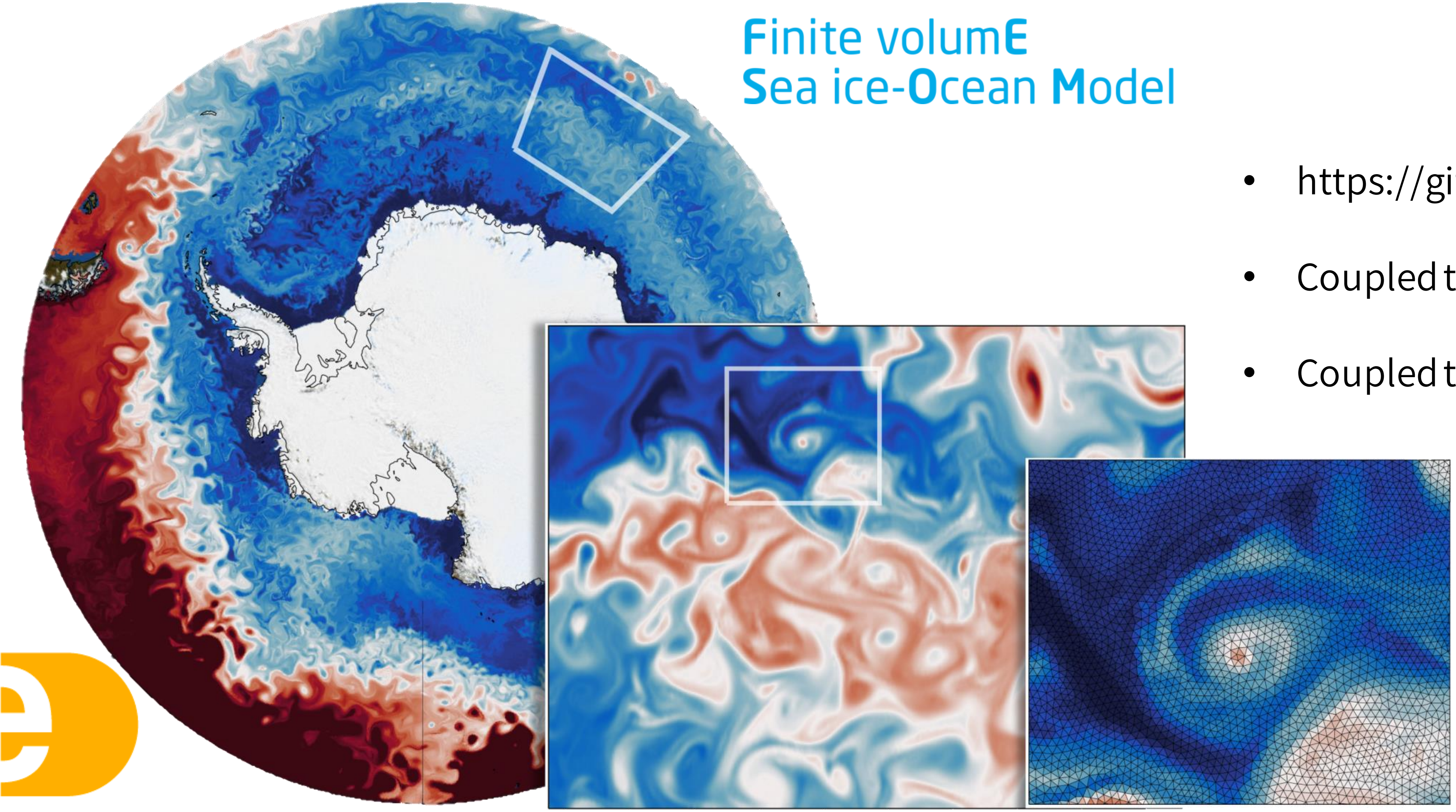
6PM from RSE's of NLeSC and ATOS offered to modeling groups around Europe:

- EMAC/MEDINA (Cyprus Institute)
- DALES (Delft University of Technology)
- RTE+RRTMGP-C++ (Wageningen University and Research)
- RegCM (Abdus Salam ICTP Trieste)
- BLOM (University of Bergen)
- AGRIF (INRIA, Grenoble)
- …
- FESOM2 (Alfred Wegener Institute, Bremerhaven)



FESOM2

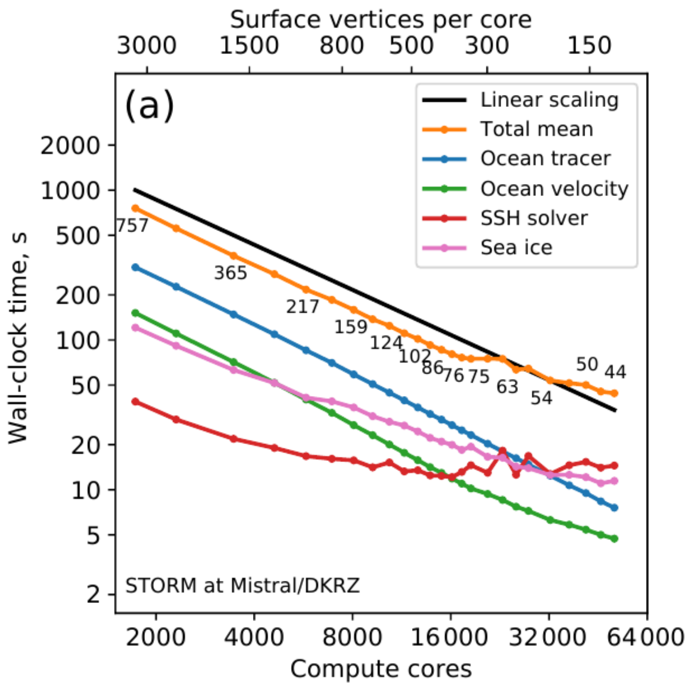
Finite volumE Sea ice-Ocean Model

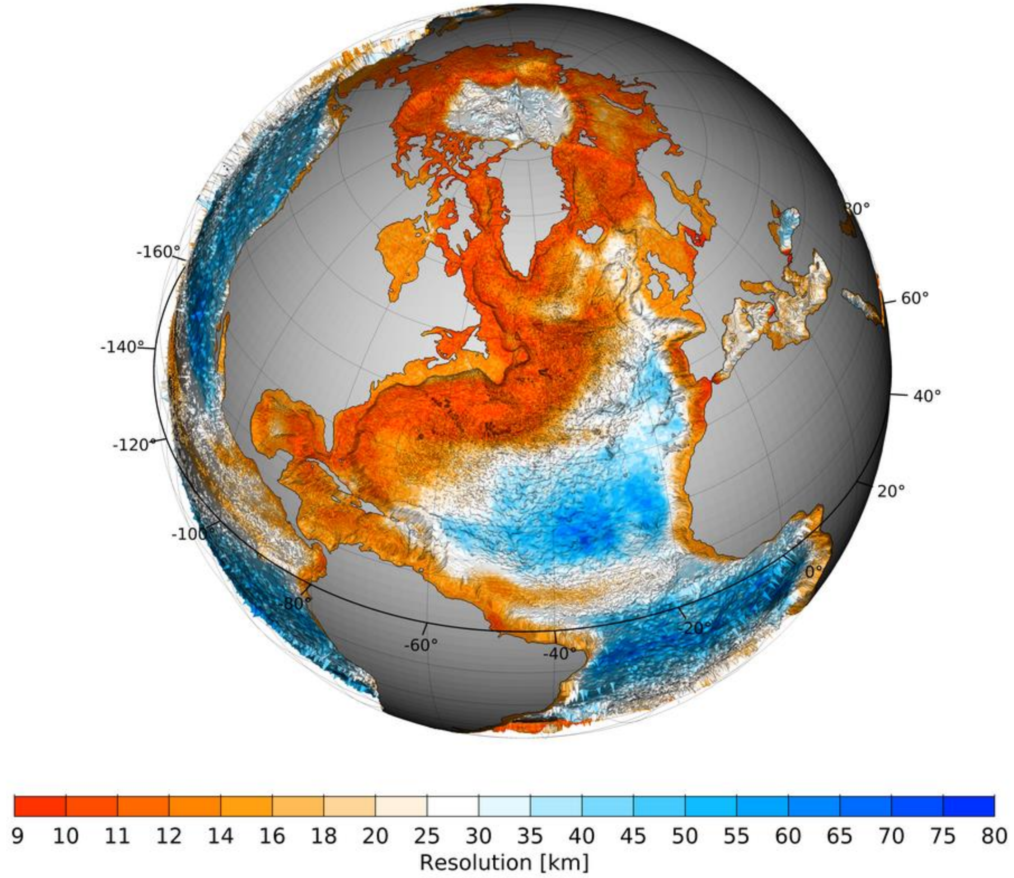


- https://github.com/FESOM/fesom2
- Coupled to ECHAM6 in AWI-CM
- Coupled to OpenIFS (NextGEMS, ...)

By N. Koldunov and D. Sein

- Unstructured mesh: local refinement
- Arbitrary Lagrangian-Eulerian vertical coordinate
- Good parallel scaling: up to ~100 mesh nodes per core



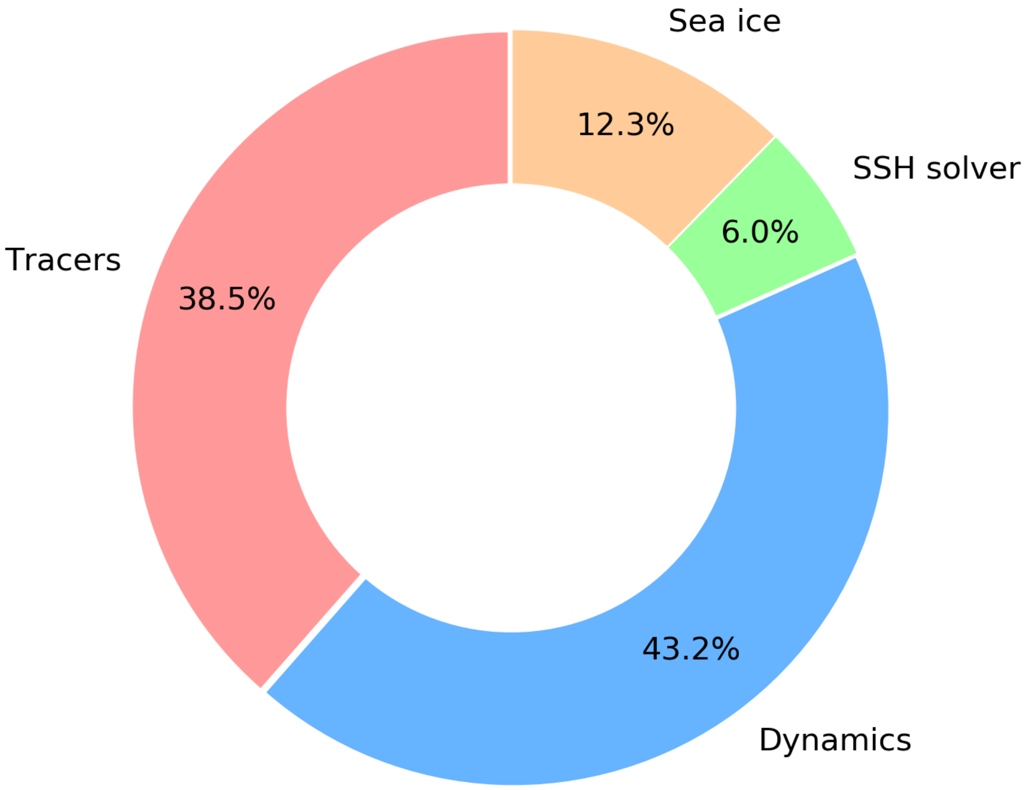


Koldunov, N., et al. "Scalability and some optimization of the Finite-volumE Sea ice–Ocean Model, Version 2.0 (FESOM2)." *GMD* 12.9 (2019): 3991-4012.

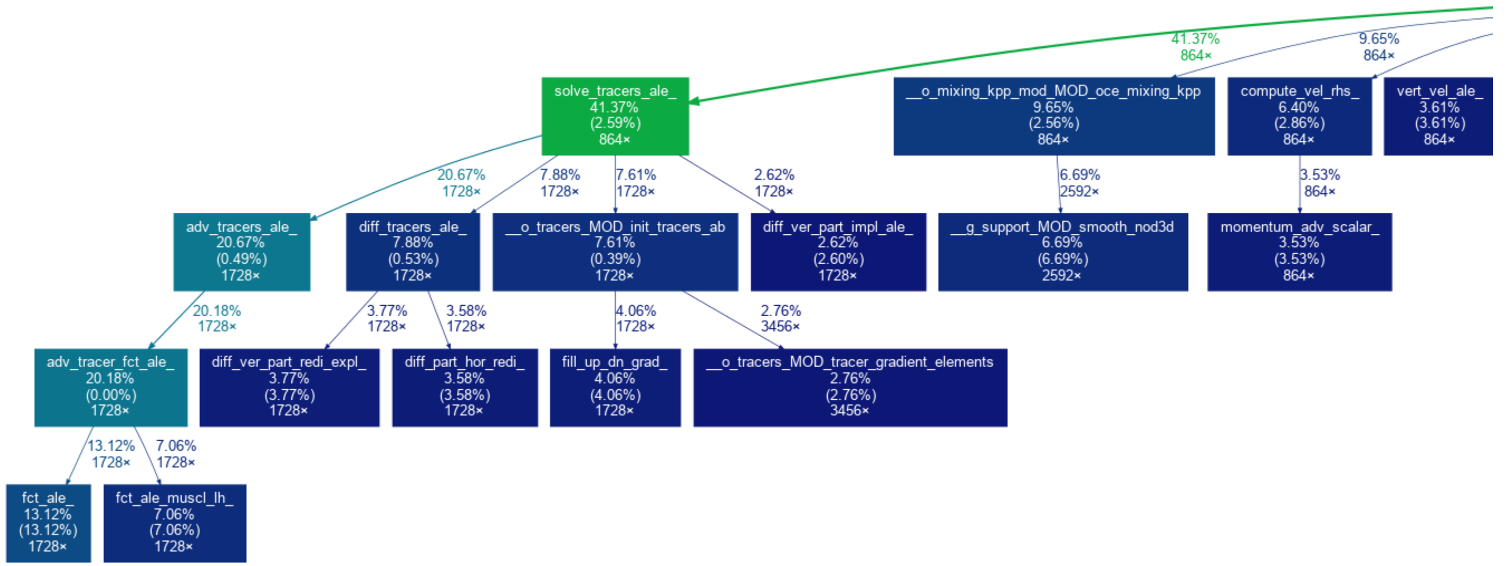
netherlands eScience center

Atos

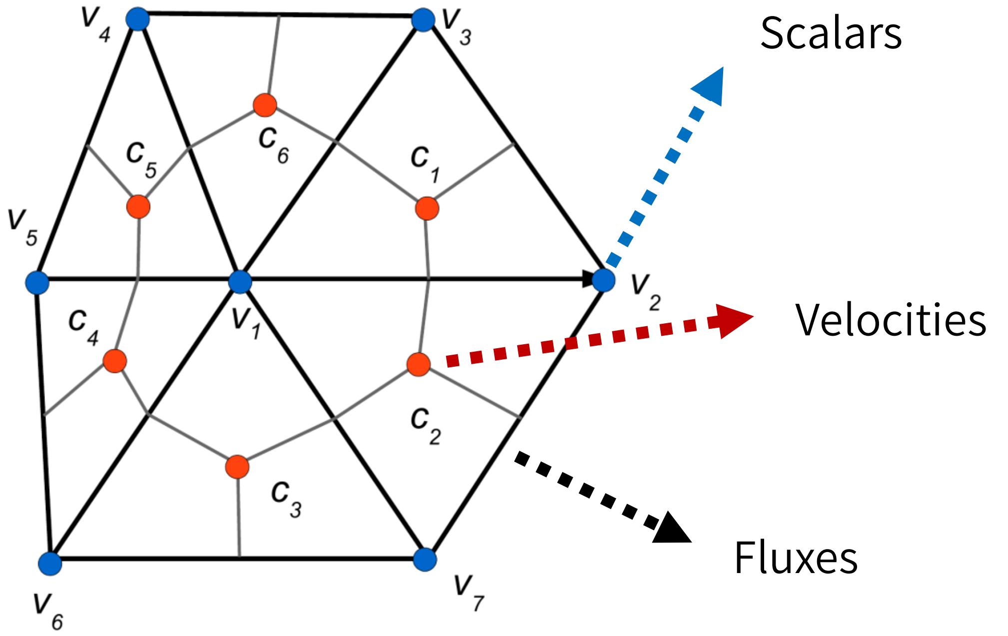
esiwace2 CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE



- Tracers: temperature and salinity sequentially
- Tracer transport slow, memory-bound, but scales well with no. MPI tasks
- Future plans: add ocean biochemistry tracers
- Flux-corrected transport slowest routine



- Variables on vertices, edges or faces (Arakawa B-grid).
- Fluxes on layer interfaces, other variables on midpoints.
- Memory layout (Fortran): (vertical, horizontal).
- Loops typically have inner z-loop, bounds depend on outer loop variable



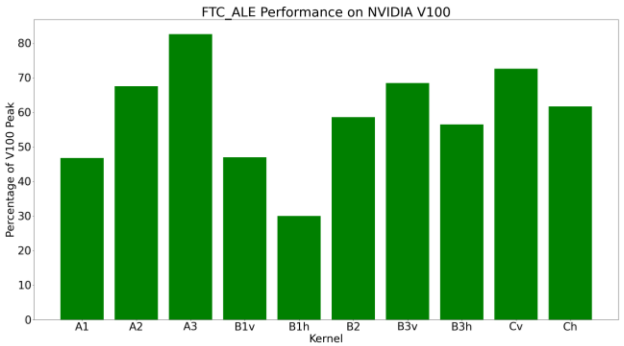
Danilov, Sergey, et al. "The finite-volume sea ice–ocean model (fesom2)." *GMD* 10.2 (2017): 765-789.

```
do n = 1, myDim_nod2D
  nu1 = ulevels_nod2D(n)
  nl1 = nlevels_nod2D(n)
  do nz = nu1, nl1 - 1
    flux = fct_plus(nz, n) * dt / areasvol(nz,n) + flux_eps
    fct_plus(nz, n) = min(1.0, fct_ttf_max(nz, n) / flux)
    flux = fct_minus(nz, n) * dt / areasvol(nz, n) - flux_eps
    fct_minus(nz, n) = min(1.0, fct_ttf_min(nz, n) / flux)
  end do
end do
```

Threads

Blocks

Flux limiting loop over vertices (`oce_adv_tra_fct` routine)



- ~20% speedup tracer transport
- Not a portable solution
- No maintenance possible from FESOM development team
- Large effort for small code section

```fortran
!$acc parallel loop gang present(…)
do n = 1, myDim_nod2D
  nu1 = ulevels_nod2D(n)
  nl1 = nlevels_nod2D(n)
  !$acc loop vector private(flux)
  do nz = nu1, nl1 - 1
    flux = fct_plus(nz, n) * dt / areasvol(nz,n) + flux_eps
    fct_plus(nz, n) = min(1.0, fct_ttf_max(nz, n) / flux)
    flux = fct_minus(nz, n) * dt / areasvol(nz, n) - flux_eps
    fct_minus(nz, n) = min(1.0, fct_ttf_min(nz, n) / flux)
  end do
end do
```

Flux limiting loop over vertices (`oce_adv_tra_fct` routine)

- ~7x speedup of `oce_adv_tra_fct` routine (no data movement)
- Somewhat portable solution.
- Can be supported by FESOM2 developers.
- In few months, full tracer transport code running on GPU's.

- Minimally intrusive port: (almost) no changes to actual Fortran code.
- Data movement between kernels minimized.
- Asynchronous kernel execution where possible.
  - Overlap MPI communication and PCIe data transfers.
  - Overlap horizontal and vertical advection/diffusion kernels.
- Explicitly tuned thread block size to 128 for Nvidia A100 GPU's
- Enable MPS daemon to mitigate context switches between MPI tasks on shared GPU's

netherlands eScience center

Atos

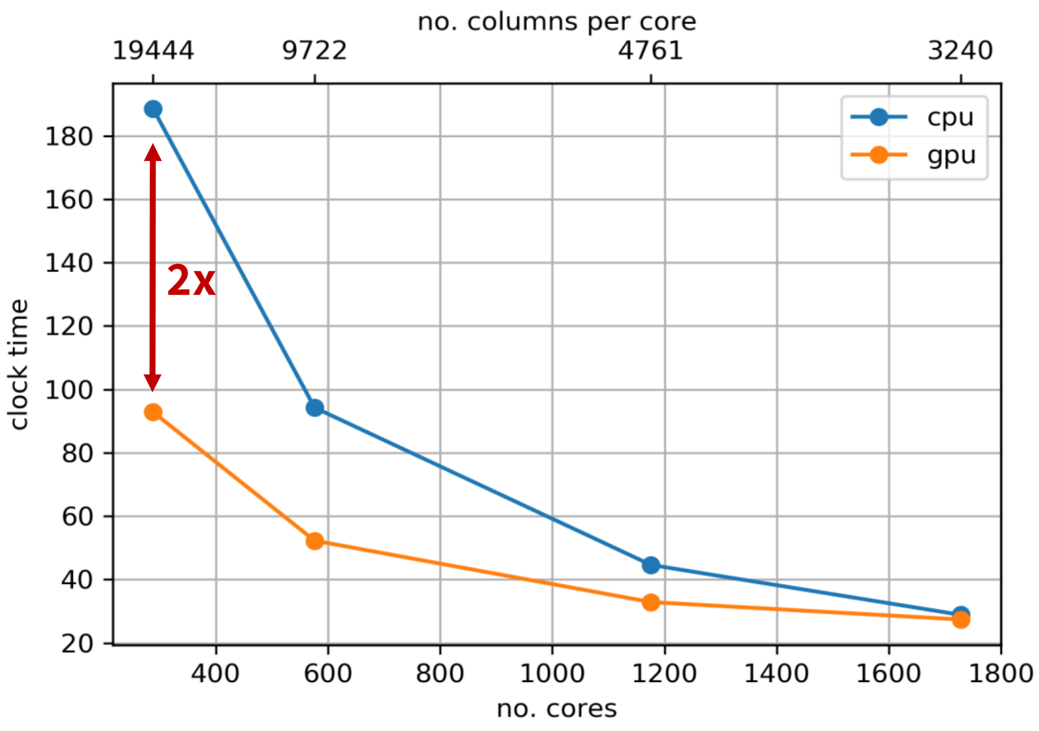
esiwace2

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

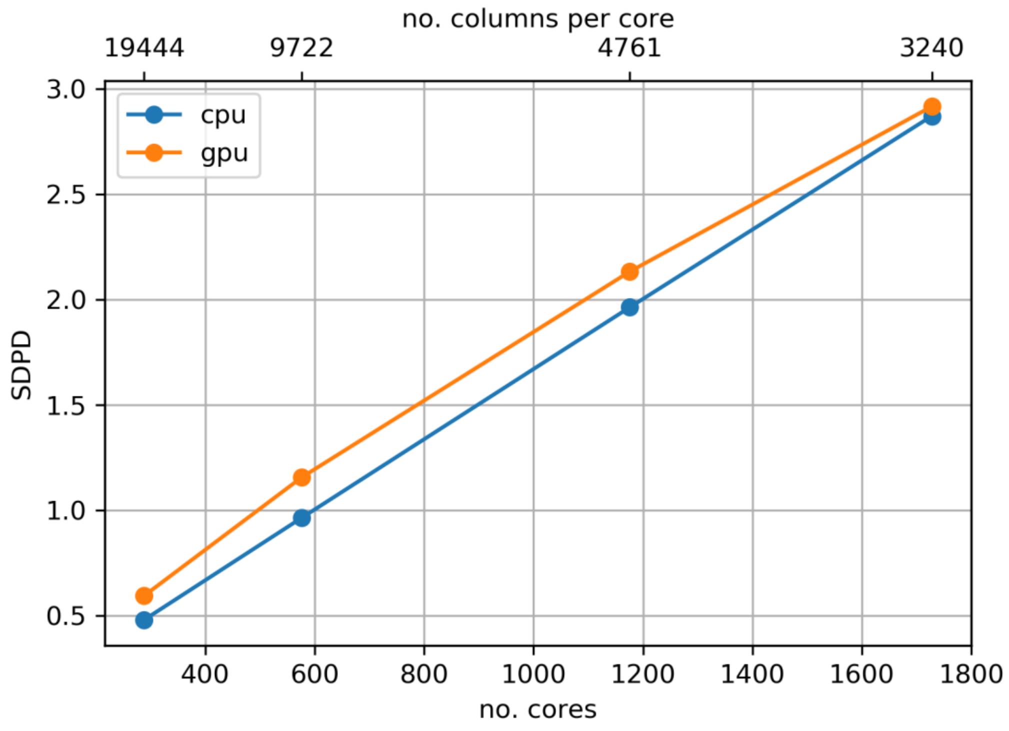
STORM test case	
Number of vertices:	5576658
Number of faces:	11095119
Resolution:	10-3 km
Rectangular analog:	0.1 degree
Number of layers:	47

JUWELS-BOOSTER	
Processor:	2 x AMD EPYC 7402 (24 cores)
GPU:	4 x Nvidia A100
Memory:	512 GB DDR4-3200
Network:	4 × Mellanox HDR200 InfiniBand
Compiler:	PGI (NVHPC) v22.3

Tracer transport



Full model



Conclusions:

- In principle, FESOM2 data structures and loops are a good fit for GPU's.
- Openacc seems the most appropriate technology to leverage GPU acceleration for FESOM2.
- Naïve openacc gives 2x speedup of tracer part for low core counts, speedup vanishes at high multiplicity.
- FESOM2 is a balanced code with many loops, partial port has limited effect

Wish list:

- Kernel optimization: leverage `!$acc collapse(2)` by replacing z-bounds with conditionals or pre-computed masks.
- Extensive validation of results.
- Extend the 'naïve' OpenACC effort to dynamics, sea ice and SSH solver
- Support CUDA-aware MPI to speed up halo exchanges in accelerated routines.
- Good solution to merge OpenACC directives/optimizations with current OpenMP parallelization efforts.

Questions