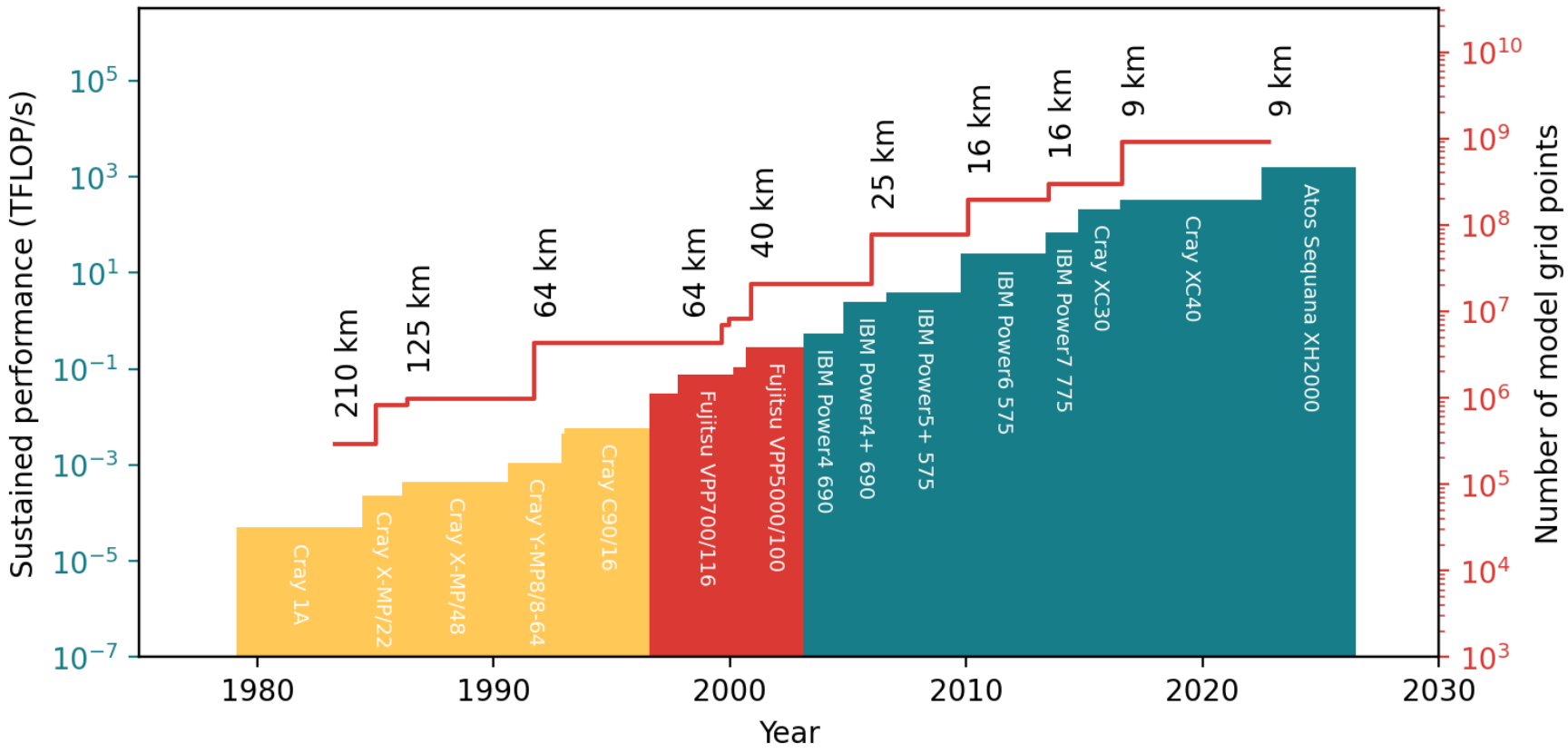
Hybrid 2024: Adapting IFS for a hybrid CPU-GPU compute model

M. Lange et al.

Michael.Lange@ecmwf.int

ECMWF

Forecast resolution is driven by sustained increase in HPC performance

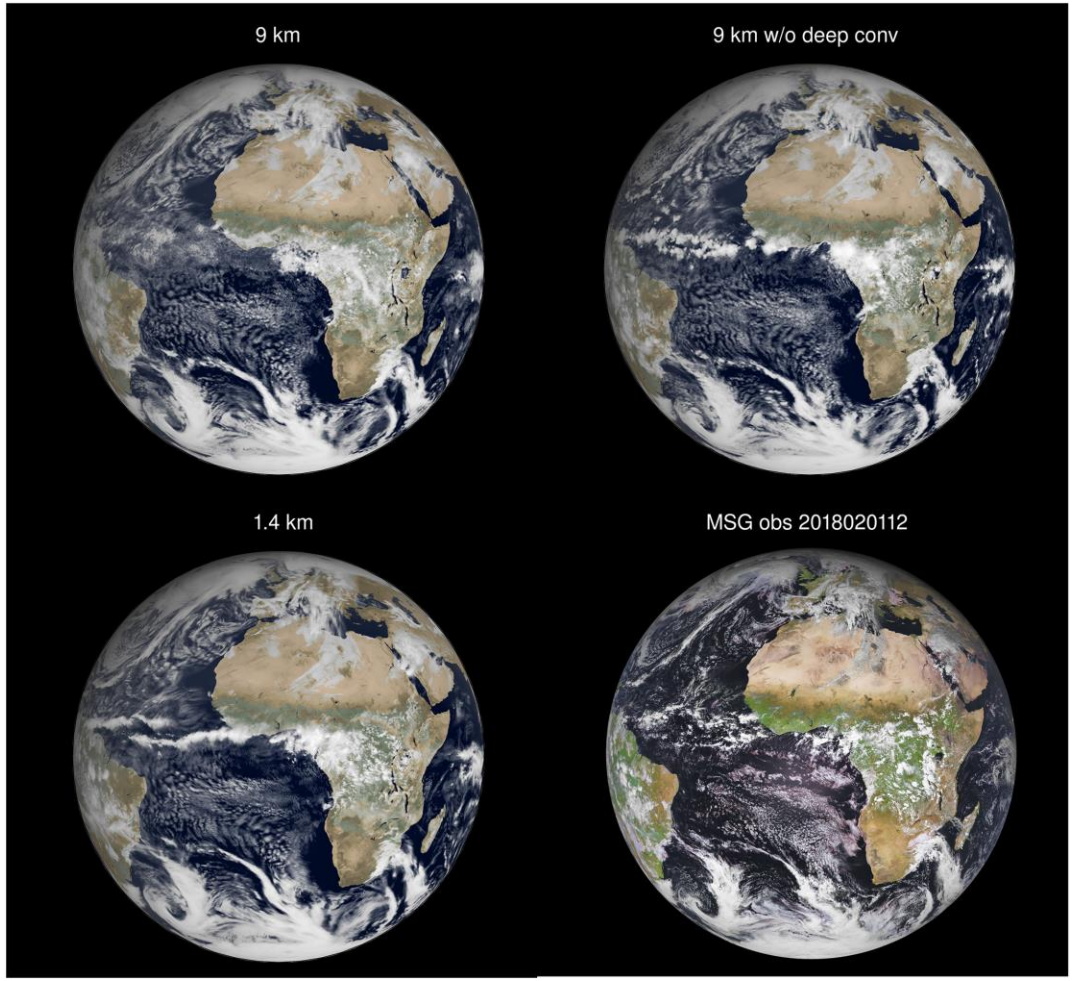


Ambitious target of 1km resolution at 1SYPD requires ~250x improvement over Cray XC40

[1] **Thomas C. Schulthess et al.** "Reflecting on the Goal and Baseline for Exascale Computing: A Roadmap Based on Weather and Climate Simulations". In: *Computing in Science & Engineering* 21.1 (Jan. 2019), pp. 30–41. DOI: 10.1109/mcse.2018.2888788.

Forecast resolution is driven by sustained increase in HPC performance

- **INCITE:** Seasonal global simulation at 1.4 km horizontal resolution
- **Destination Earth:** EC initiative to develop highly accurate digital twin of Earth
- **EuroHPC:** Access to large-scale computing platforms via Destination Earth
- Many (Pre-)Exascale-class systems feature large GPU partitions



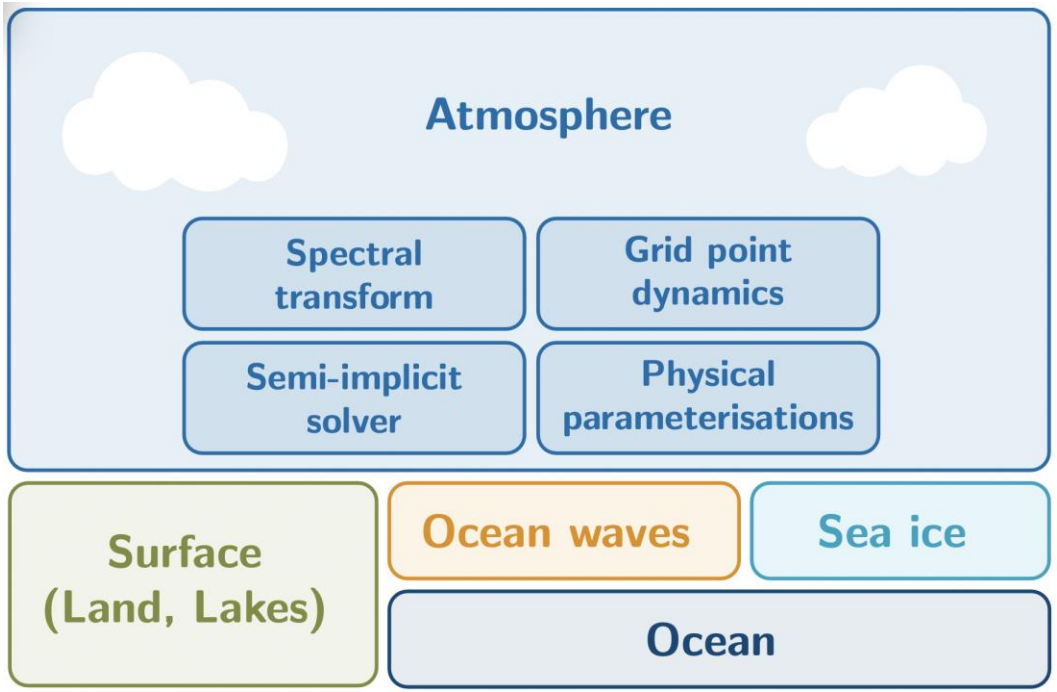
[2] **Nils P. Wedi et al.** "A Baseline for Global Weather and Climate Simulations at 1 km Resolution". In: *Journal of Advances in Modeling Earth Systems* 12.11 (2020), e2020MS002192. DOI: https://doi.org/10.1029/2020MS002192.

Hybrid 2024 - Preparing IFS for HPC accelerators

Hybrid 2024: Core project to adapt IFS to accelerator-based HPC architectures

Accelerator-enabled multi-architecture IFS

- Sustainable technical development of accelerator capabilities alongside scientific development
- Incremental adaptation of model components to different accelerators and programming models
- Dedicated build-modes and testing for different hardware architectures to assess capabilities
  - GPU architectures are primary focus
  - Alternative CPU architectures and accelerators are also considered

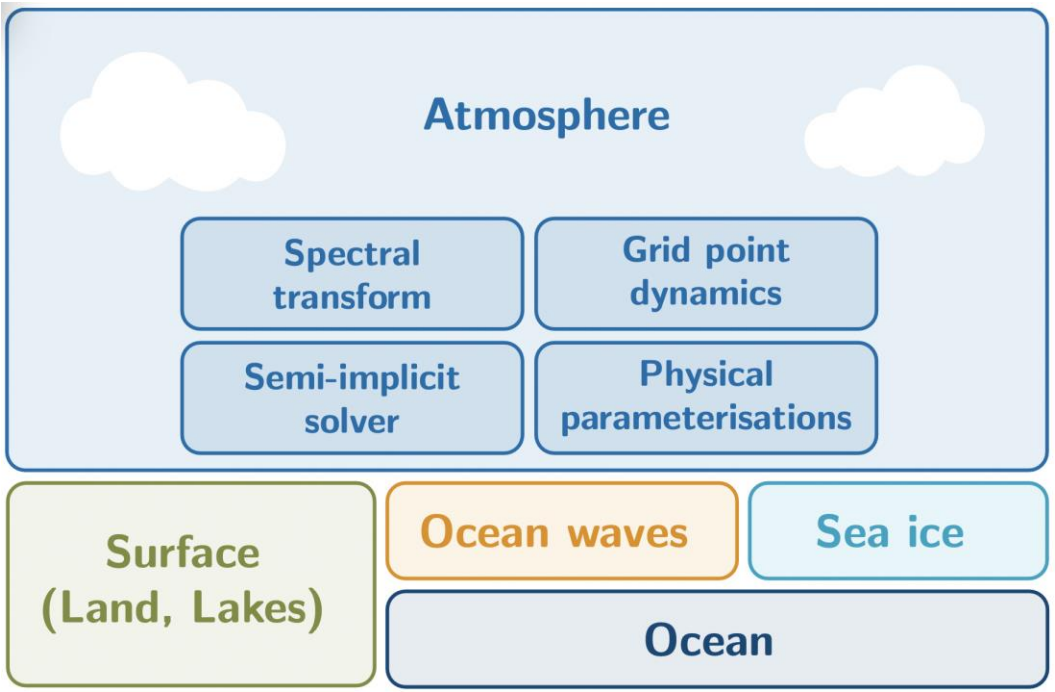


Hybrid 2024 - Preparing IFS for HPC accelerators

The challenge: Adapt IFS to new programming paradigms without harming CPU performance

Multiple programming models in a single code

- Use library APIs to hide technical complexities
- Use flexible data structures to deal with complex memory hierarchies
- Increasingly flexible control flow with different parallelisation paradigms
- Source-to-source translation for device-specific optimisation of compute kernels



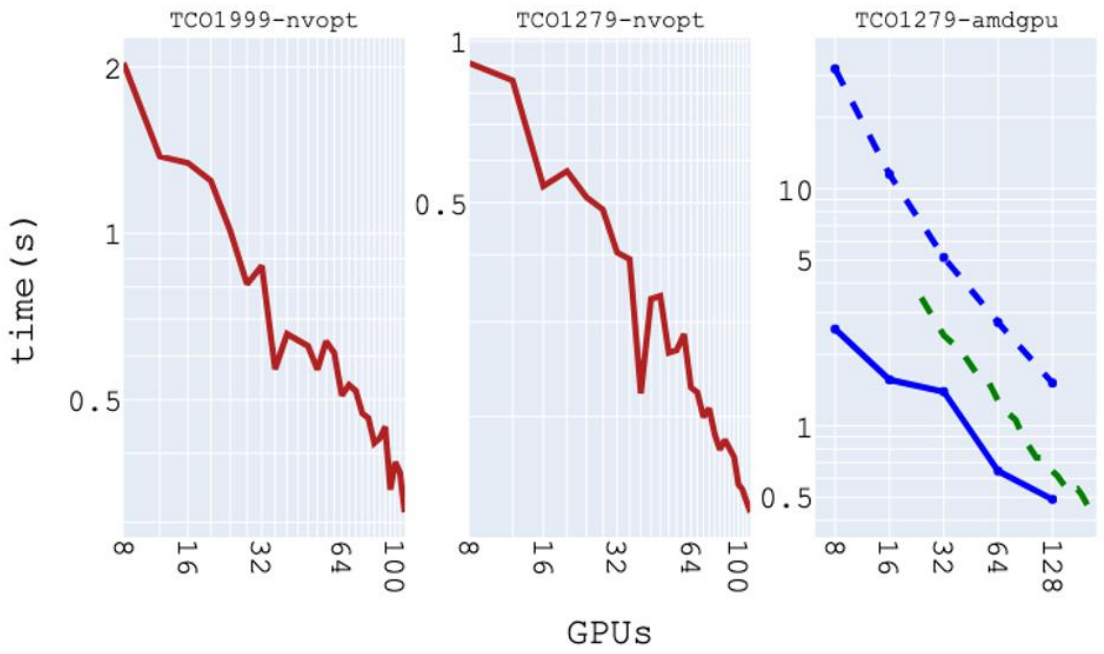
Libraries and standalone components

ecTrans - An open-source Spectral Transform library

- Extracted Spectral Transform routines and packaged as standalone project (github.com/ecmwf-ifs/ectrans)
- Integrated into mainline IFS; dependencies via standalone FIAT library (github.com/ecmwf-ifs/fiat)
- Standalone mode is used to assess multi-node scaling behaviour

GPU-enabled version available

- Nvidia and AMD support (red-green) in branch
- Good scaling behaviour with CUDA-aware MPI (GPU-to-GPU MPI communications via NVLink)
- **More performance optimisations on the way!**
- **WARNING: This is work in progress! Not all features are supported on GPU yet!**



GPU-enabled data structures

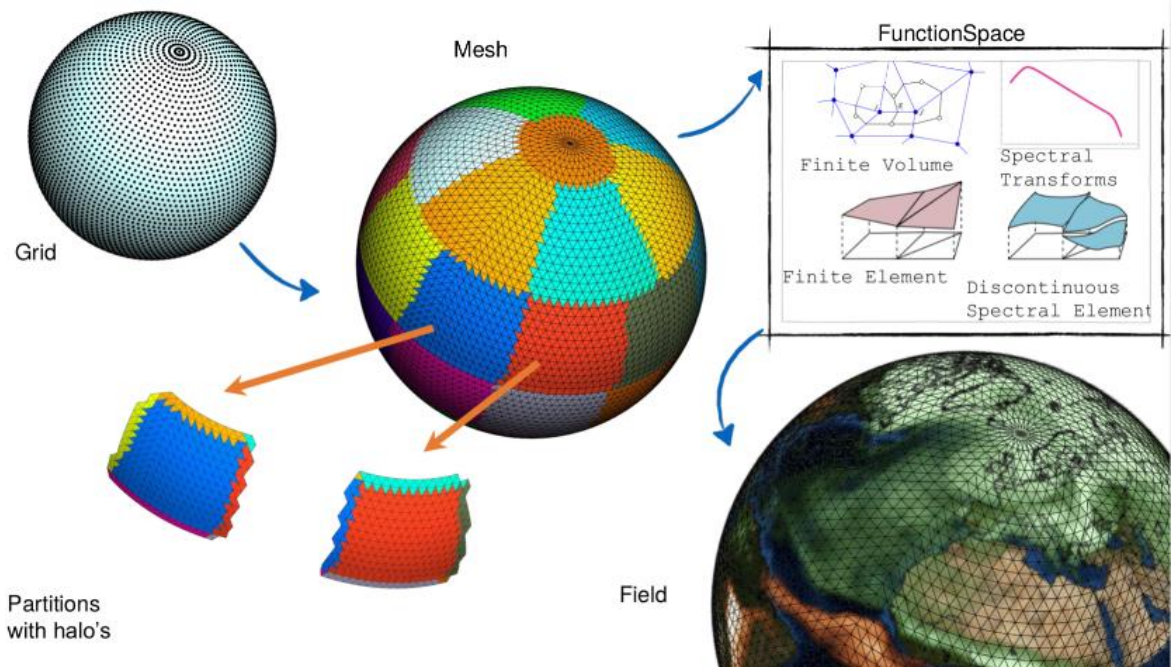
Flexible data structures for complex memory hierachies

FIELD API: Initial adaptation to allow GPU-offload via OpenACC / OpenMP

- **Object-oriented** data structures to encapsulate memory placement of field arrays
- **Separation of concerns:** Explicitly manage data offload to accelerator devices
- Enables restructuring of control flow to adapt to alternative execution modes

Atlas – A library for NWP and climate modelling

- Modern C++ library with Fortran interfaces
- Data structures for numerical algorithms:
  - Remapping and interpolation
  - Gradient, divergence, laplacian
  - Spherical Harmonics transforms
  - **Increasing accelerator-awareness!**

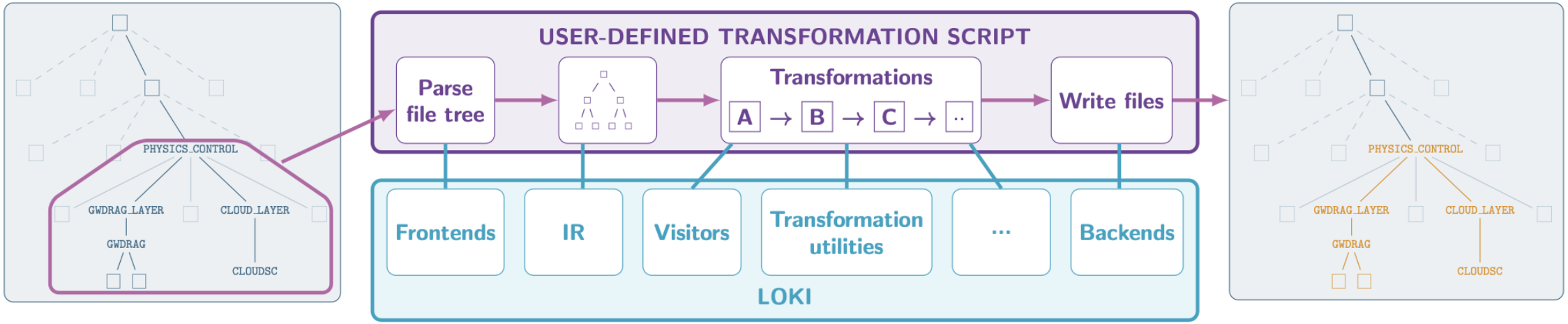


[3] **Willem Deconinck et al.** "Atlas : A library for numerical weather prediction and climate modelling". In: *Computer Physics Communications* 220 (2017), pp. 188–204. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2017.07.006.

Source-to-source transformations

Source-to-source translation of physical parametrisations

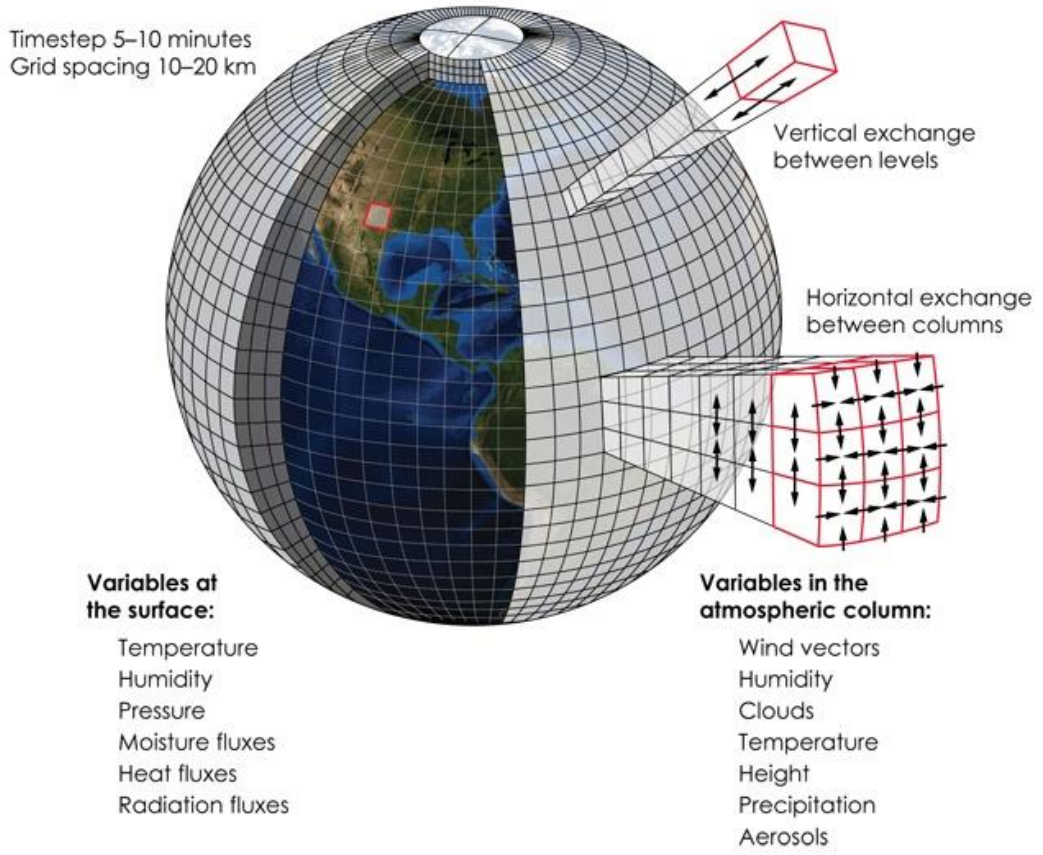
- **Loki:** Programmable source-to-source translation paclage written in Python
  - Library of tools and APIs to build your own source transformation recipes
  - Built on basic principles of compiler technology: IR trees, visitors and transformers
- Enables **batch transformation of source tree** at compile time (“complex preprocessor”)
- **Freely programmable:** Development of recipes (eg. for GPU) is **driven by experts!**



Source-to-source translation of physical parametrisations

- No data dependencies between columns: Lots of parallelism!
- Scientific kernel can be developed and tested for a single column
- In IFS columns are stored in a block layout traversed with a high OpenMP loop

- IFS-specific code transformation recipes
  - Loki-SCC: Single Column Coalesced
  - Loki-SCCH: SCC with argument hoisting
  - **Make specific use of IFS data layout!**



[4] **Valentin Clement et al.** "The CLAW DSL: Abstractions for Performance Portable Weather and Climate Models". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '18. 2018. ISBN: 9781450358910. DOI: 10.1145/3218176.3218226.

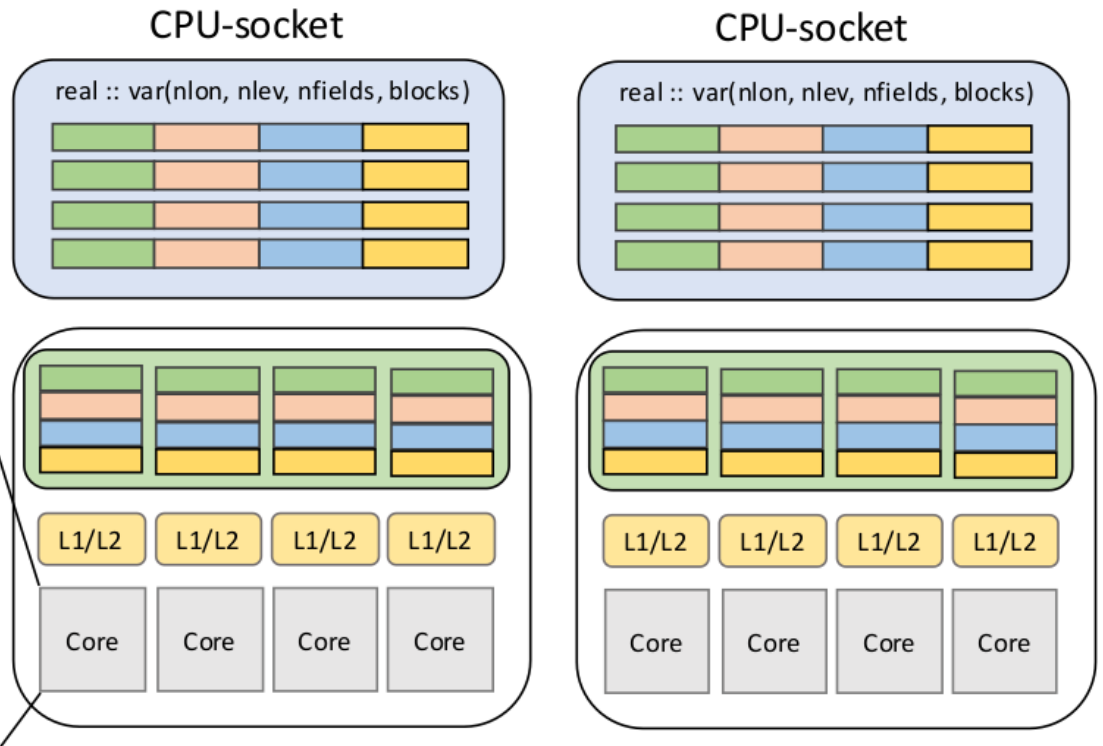
IFS: Memory data layout and parallelisation

```
!$omp parallel loop
do ibl=1, nblocks
   call kernel(var1(:,:,ibl), var2(:,ibl), …)
end do
```

```
SUBROUTINE KERNEL(nlon,nlev,var1,var2,…)
  real :: var1(nlon, nlev)
  real :: var2(nlon)

  do j=1, klon
    var1(j, 1) = var2(j)
  end do

  do k=2, nlev
    do j=1, klon
      var1(j,k) = var1(j,k-1) + <update>
    end do
  end do
END SUBROUTINE
```



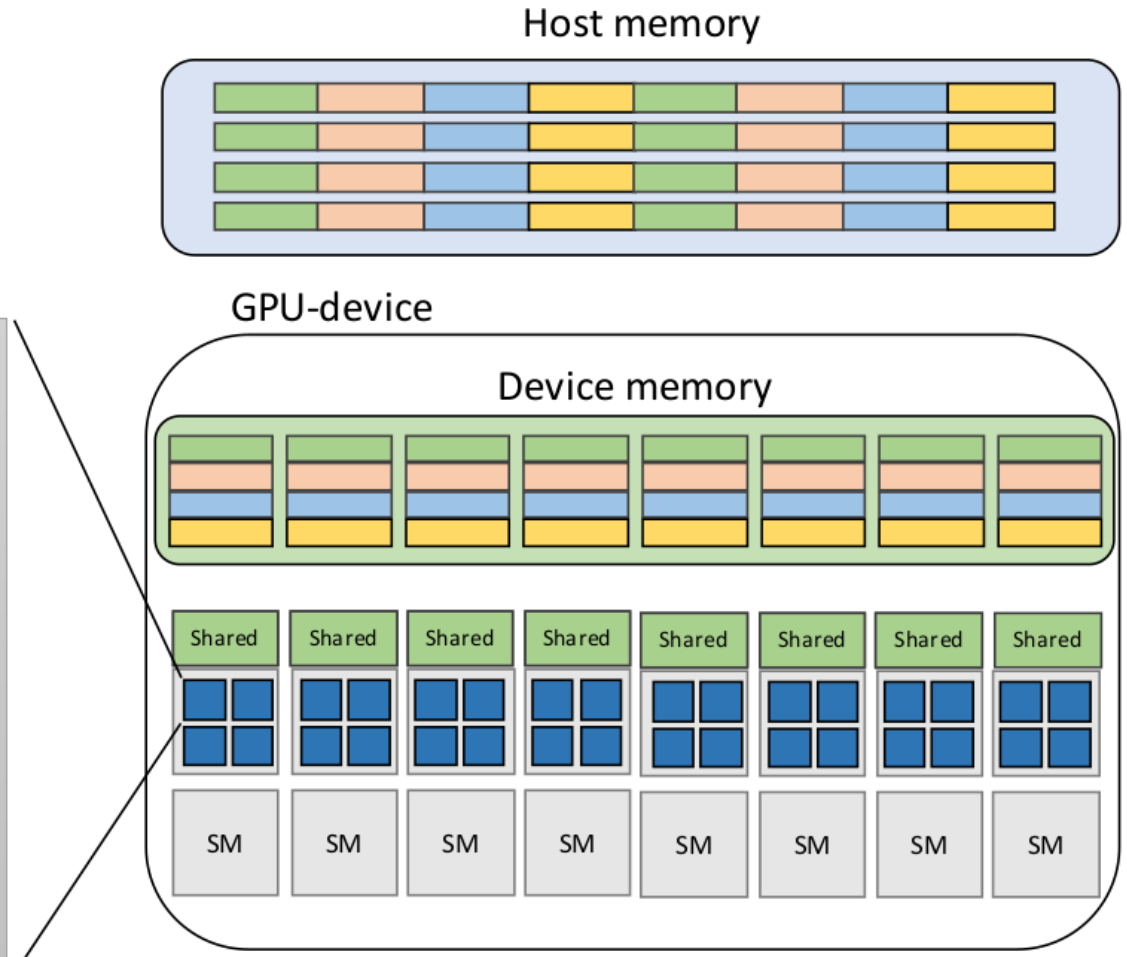
IFS: Memory data layout and parallelisation

```
!$acc parallel loop gang
do ibl=1, nblocks
   call kernel(var1(:,:,ibl), var2(:,ibl), …)
end do
```

```
SUBROUTINE KERNEL(nlon,nlev,var1,var2,…)
  real :: var1(nlon, nlev)
  real :: var2(nlon)
!$acc routine vector

  !$acc loop vector
  do j=1, klon
    var1(j, 1) = var2(j)

    !$acc loop seq
    do k=2, nlev
      var1(j,k) = var1(j,k-1) + <update>
    end do
  end do
END SUBROUTINE
```

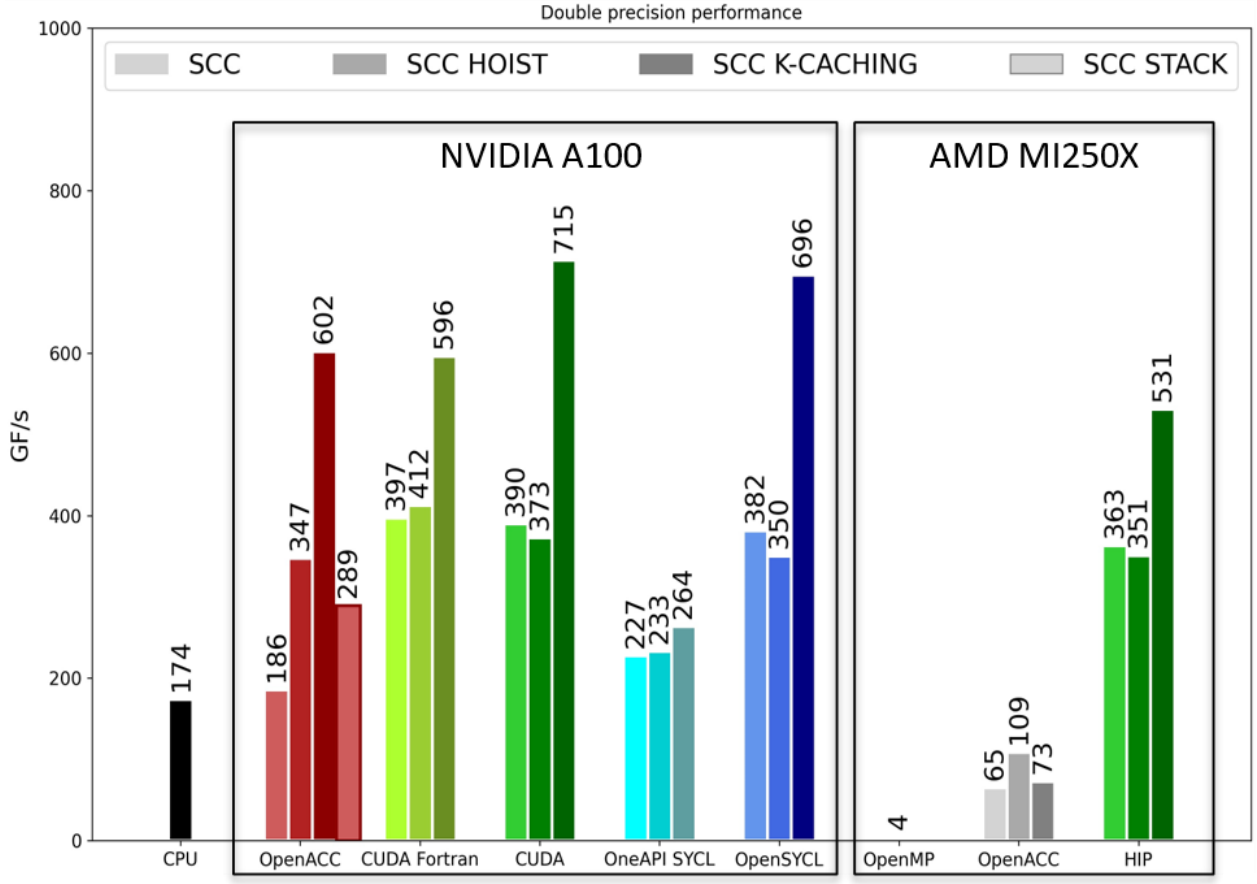


Source-to-source translation of physical parametrisations

- **CLOUDSC** (github.com/ecmwf-ifs/dwarf-p-cloudsc)
  - Representative parallelisation, memory layout
  - Challenging to optimise (high register pressure)

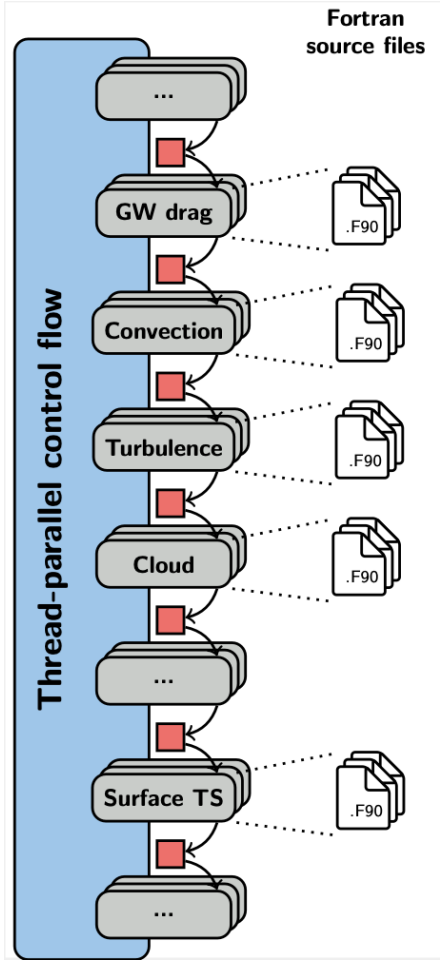
- **Evaluation of GPU code transformations**
  - Compiler and programming model evaluation
  - Per-chip/socket performance comparison
  - Comparison of different compilers, programming models, loop strategies and memory management

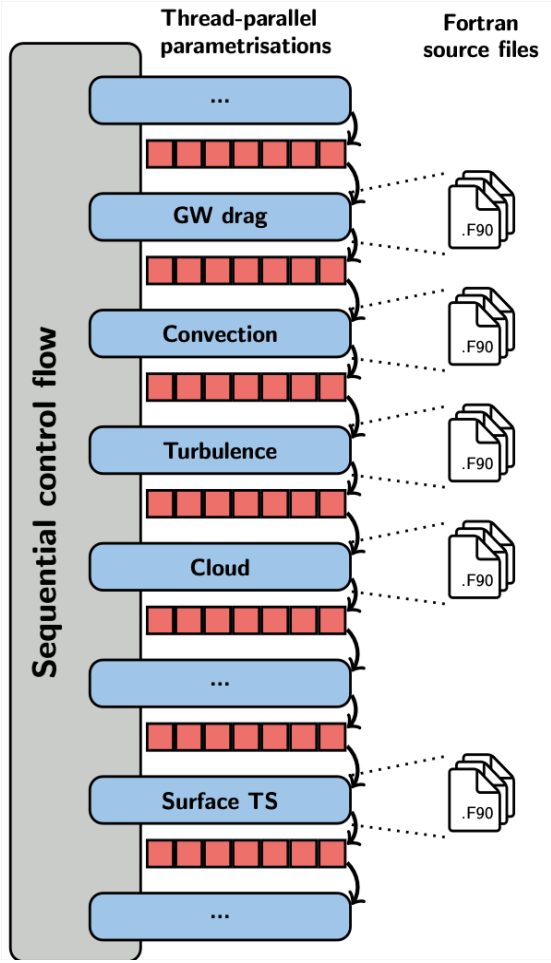
- **Development of Loki-SCC recipe family**
  - Loop flip and parallelization for “gang-vector” mode
  - Different methods for dealing with temporary arrays
    - Vector / Hoist / Stack / K-cache / …



Adapting EC-Physics to GPU

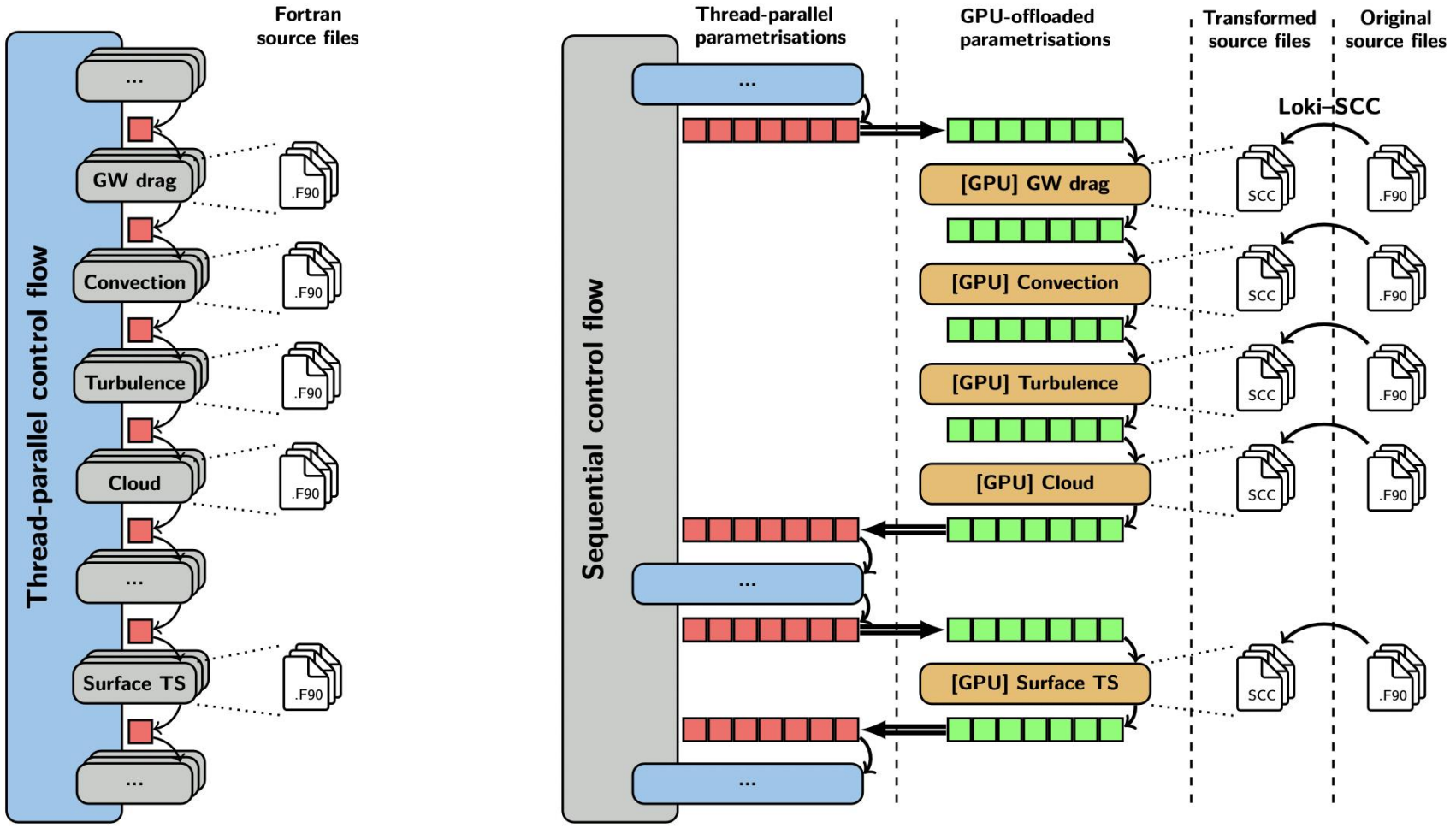
EC-Physics prototype - Control flow restructuring





- IFS-specific parallelisation pattern
  - Single large thread-loop
  - Many separate components
  - Hard to debug and develop!
- FIELD API array abstraction
  - IFS-specific data layout
  - Many(!) temporary arrays between and within parametrisations
  - Data scope changes to "buffer" data between separate kernels

EC-Physics prototype - Control flow restructuring

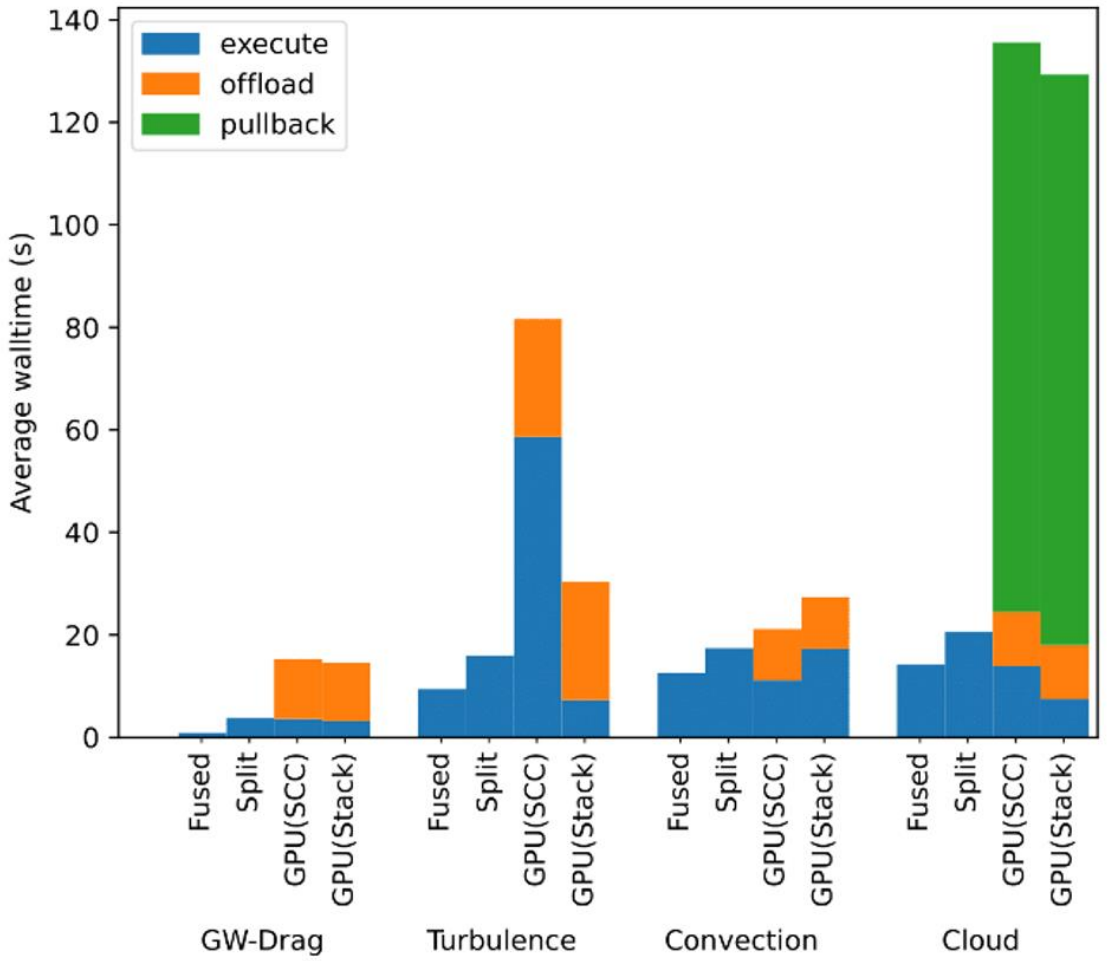


EC-Physics prototype - Performance

Demo on tco399 on AC GPU partition (Bologna)

- FIELD API use to offload and “pull back” data
  - Pullback not yet optimized
  - Data transfer can be improved with CUDA
- Different SCC-family recipes
  - Loki-SCC-stack (thank you, Météo-France!)
  - Many other fixes over previous examples
- Compute performance beating CPU
  - Still artificial configurations

- **GPU adaptation is ongoing...**
  Alongside science developments!



Takeaway messages

- **Hybrid 2024** - Sustainable accelerator (GPU) adaptation alongside scientific development
  - Close collaboration with member state and strong synergies with Destination Earth

- **Focus on refactoring and software infrastructure**
  - Increased modularisation and use of library interfaces
  - GPU-enabled data structures and preparation towards Atlas
  - Source-to-source code transformation via IFS-specific recipes

- **Enable continuous adaption and performance optimisation**
  - Monitor and prepare for hardware trends outside of procurement cycle
  - Maintain compatibility with multiple HPC architectures (Destination Earth)

Thank you! Any questions?

michael.lange@ecmwf.int
MLange805