GPU adaptation of the IFS

M. Lange, A. Nawab, B. Reuter, J. Ericsson, M. Staneker, O. Marsden,

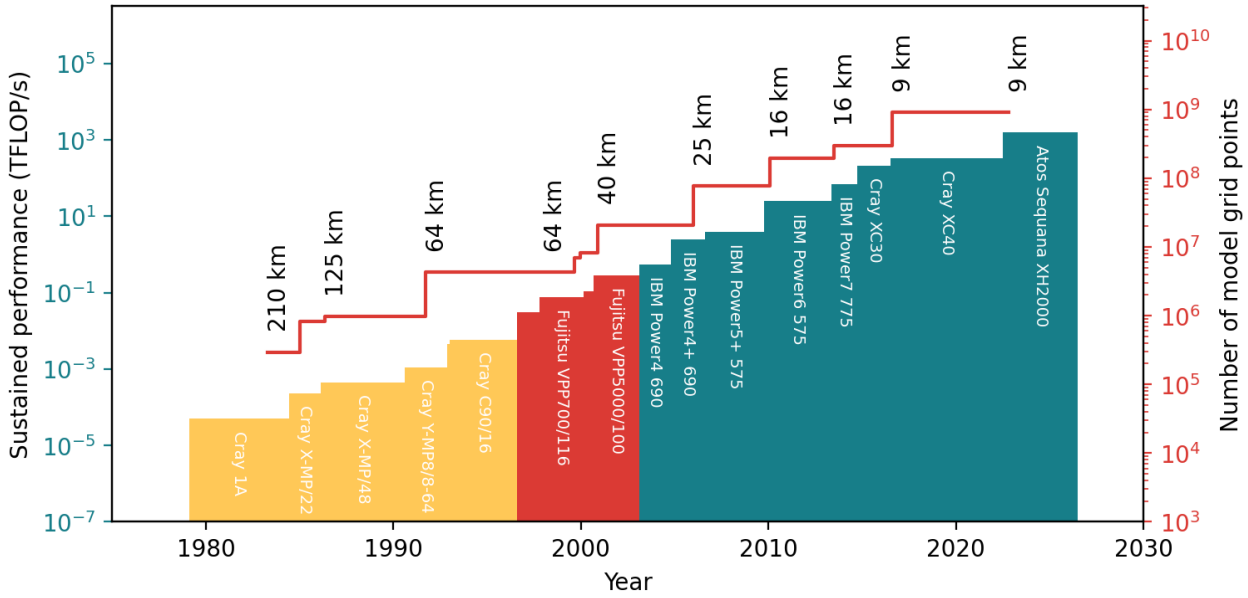
P. Gillies, Z. Piotrowski, F. Pouyan

Michael.Lange@ecmwf.int

ECMWF

The ECMWF HPC facility

- **Operational HPC facility since 1979**
  - Initially Reading, Bologna since 2022
- **Steady increase in compute power** (Moore's Law) has driven a steady increase in resolution
  - *Computing paradigm has changed but not often…*
- **And now we have GPUs and AIFS!**
  - Small partition of dual-CPU, quad-A100 nodes
  - New partition of 4-way Grace-Hopper nodes

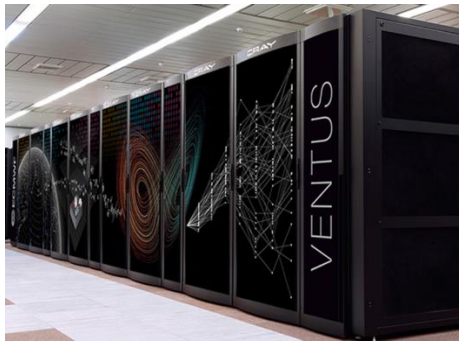


Cray 1A

Cray X-MP/22

Fujitsu VPP700

IBM Power4 690



Cray XC30

Atos Sequana XH2000

Running IFS on EuroHPC and external systems

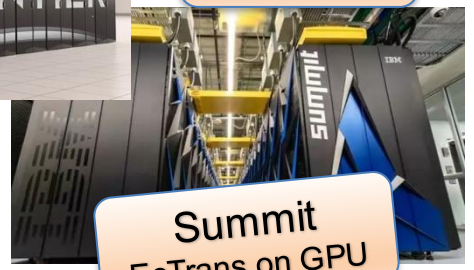
LUMI-C
Extremes DT (IFS-NEMO)
Climate DT (IFS-FESOM)

Destination Earth | EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

EuroHPC
Joint Undertaking

Fugaku
IFS on ARM

Frontier
IFS @ 1.4km
EcTrans on GPU



Summit
EcTrans on GPU

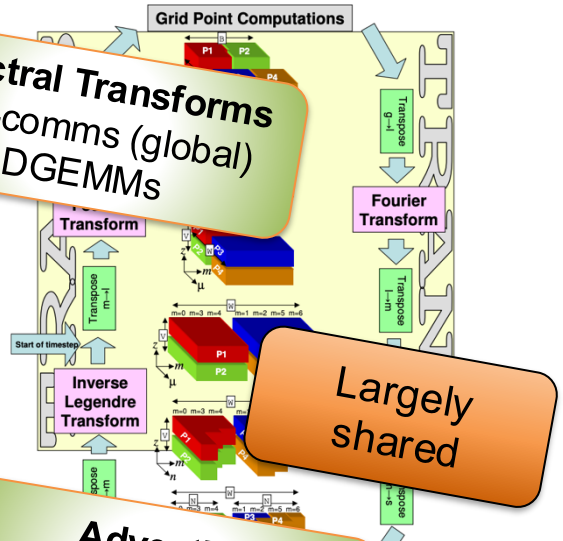
JEDI / JUPITER
IFS-NEMO
IFS-RAPS (GPU)

MN5-GPP
Climate DT
IFS-FESOM

Leonardo-DCGP
Extremes DT
IFS-NEMO

The challenge of adapting the IFS forecast model (to GPU)

Dynamical core



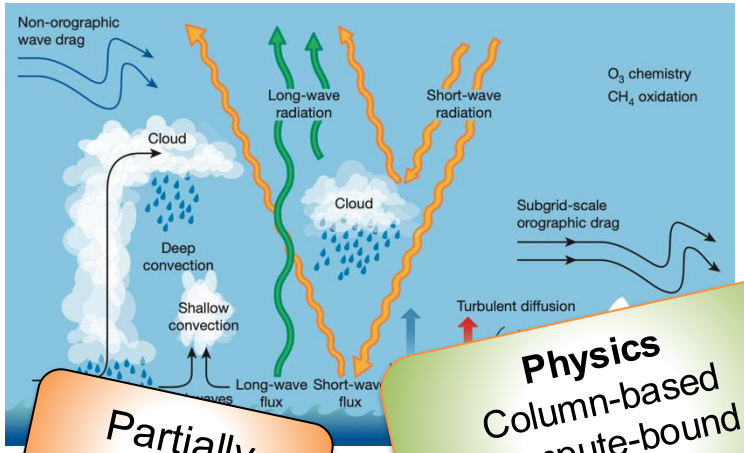
Spectral Transforms
MPI-comms (global)
DGEMMs

Largely shared

Advection
MPI-comms (local)
Memory-bandwidth

- **Large Fortran code base**
  - ~ 1 MLoC for forecast
  - > 5 MLoC, 15k files total (with DA)
- **Many computational patterns**
  - No overarching abstraction
  - No single optimisation strategy
- **Shared code and data structures**
  - Météo-France and ACCORD
  - Shared underlying data model

Physical parametrisations



Partially shared

Physics
Column-based
Compute-bound
No MPI-comms

Ocean Model

NEMO

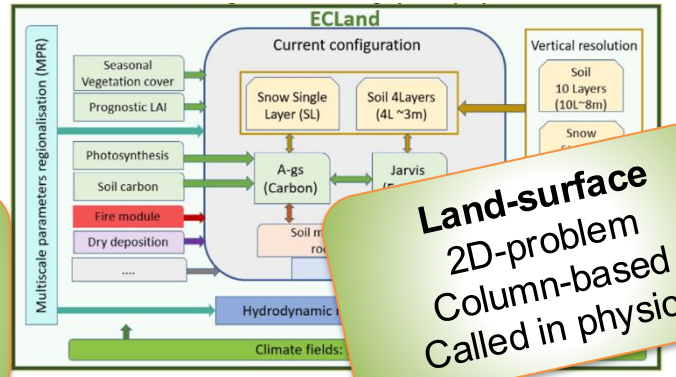
External Code

NEMO
Low-order stencil
Memory-bandwidth
MPI-comms (local)

Wave Model

Wave
Horizontal stencil
Column-based
MPI-comms (local)

Land surface model



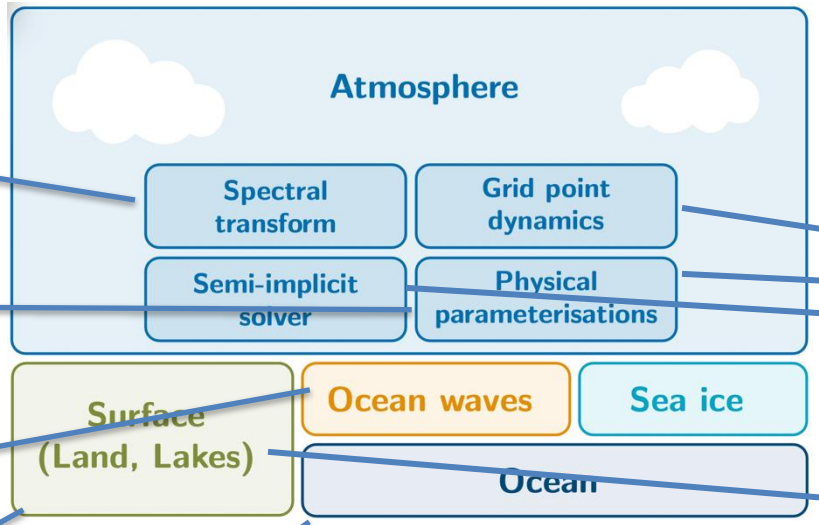
Land-surface
2D-problem
Column-based
Called in physics

Preparing IFS for GPUs and multiple HPC systems

Modularisation

- ecTrans — Spectral Transform
- ecRad — Radiation model
- ecWAM — Wave model
- ecLand — Land surface model

PSyclone NEMO

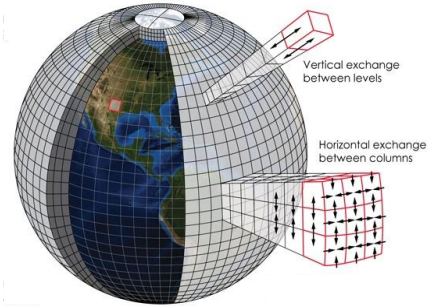


GPU-enabled data structures

Generalised data transfer API

- Atlas
- FIELD API

Source-to-source



Single-column recipes

Optimise compute kernels

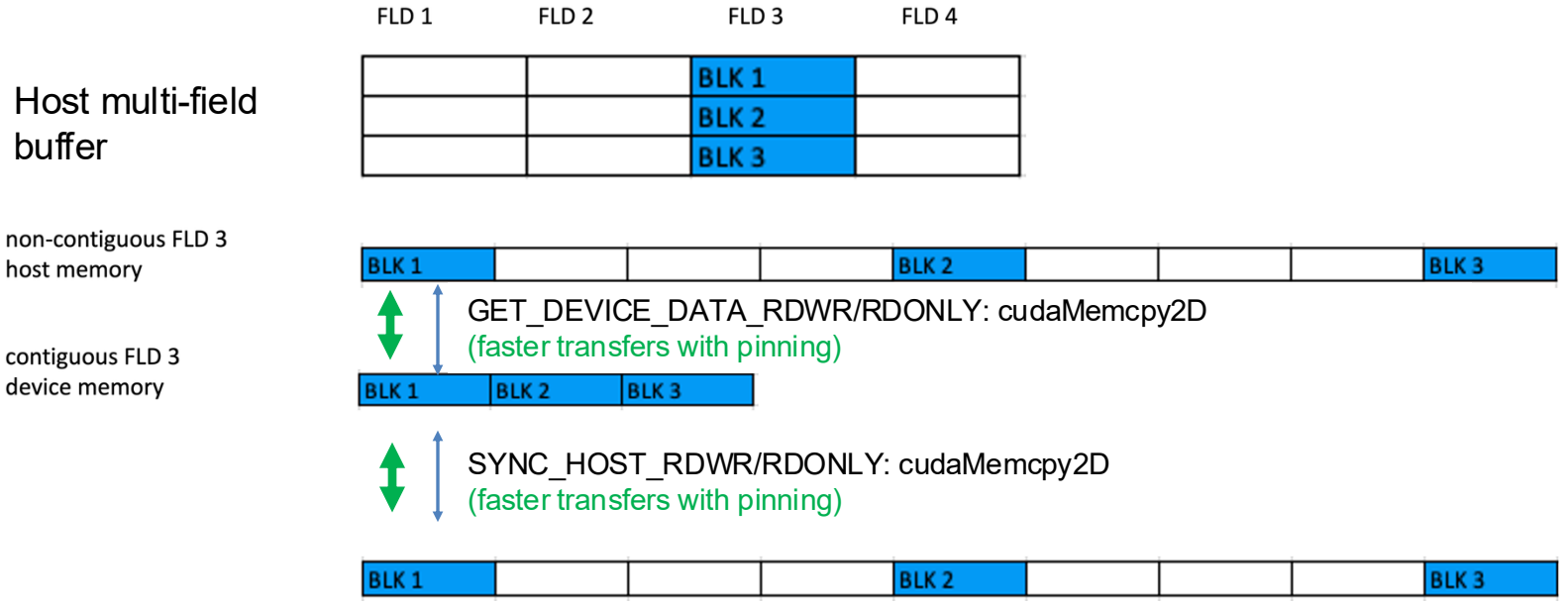
Schedule data transfers

Loki S2S tool

Flexible data structures for complex memory hierarchies

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

- **Jointly developed** by Météo-France and ECMWF
- **Object-oriented** data structures to encapsulate memory placement of field arrays
- **GPU support:** Vendor-agnostic data offload to GPU devices
- **GPU optimization:** Dedicated CUDA backend for faster transfers and memory pinning
- **For more, see poster by J. Ericsson!**

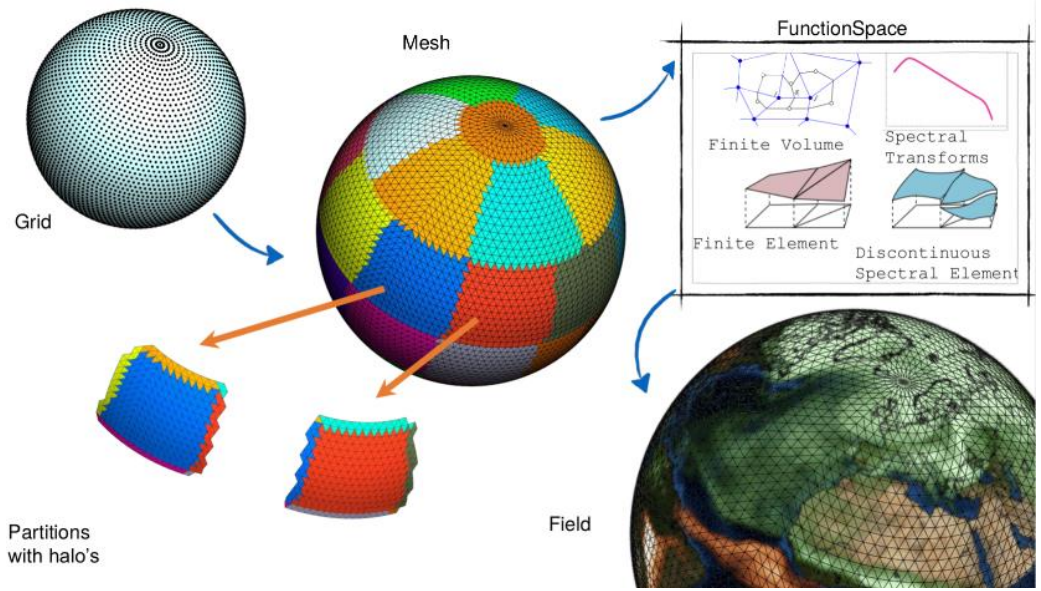



```
REAL :: BUFFER(NPROMA,4,NBLKS)
REAL, CONTIGUOUS, POINTER :: D_PTR(:,:)
CLASS(FIELD_2RB), POINTER :: F

CALL FIELD_NEW(F, DATA=BUFFER(:,3,:))

CALL F%GET_DEVICE_DATA_RDWR(D_PTR)

[compute on device using D_PTR]

CALL F%SYNC_HOST_RDWR()
```

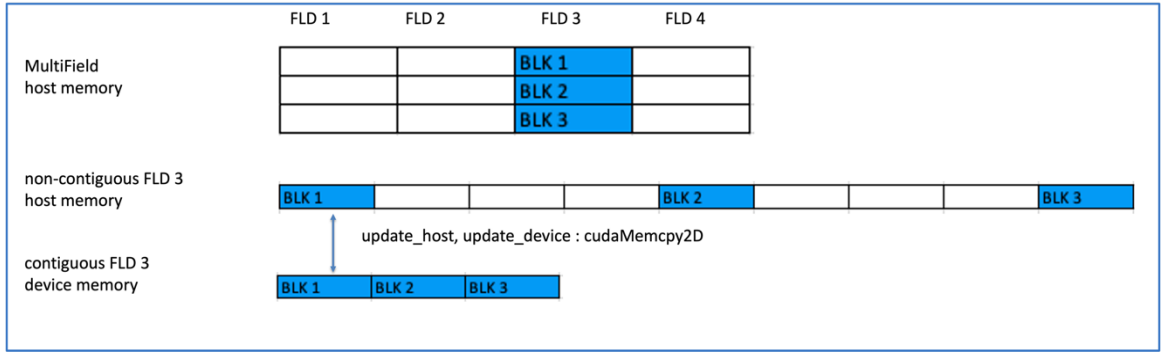
Flexible data structures for complex memory hierarchies

Atlas – A library for NWP and climate modelling [1]

- Modern C++ library with Fortran interfaces
- Data structures library for numerical algorithms
  - Interpolation, remapping, spherical harmonics
  - Gradient, divergence, Laplacian operators



- **Increasing GPU-awareness!**
  - CUDA enabled storage and data transfers
  - IFS-style memory blocking for host/device

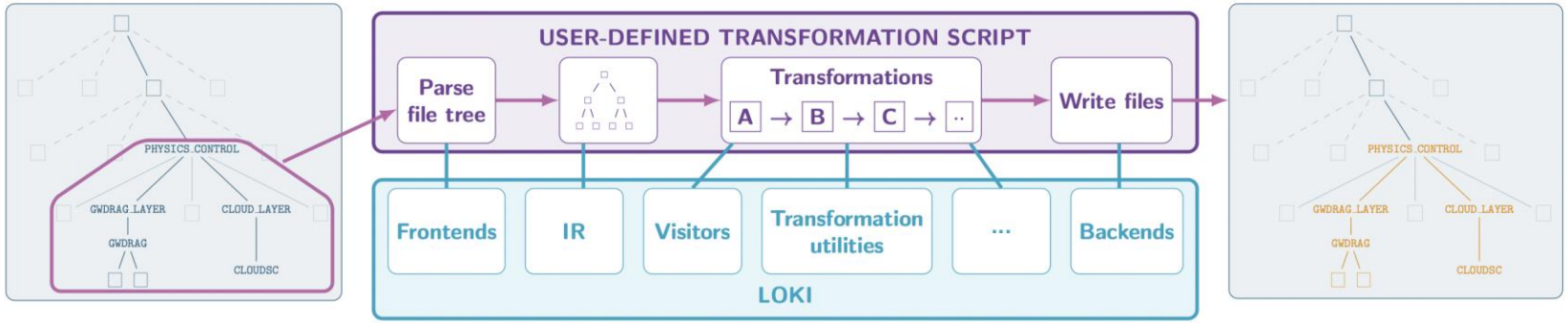


See later talk "**Flexible GPU offloading strategies with the Atlas library using Pluto**" by W. Deconinck

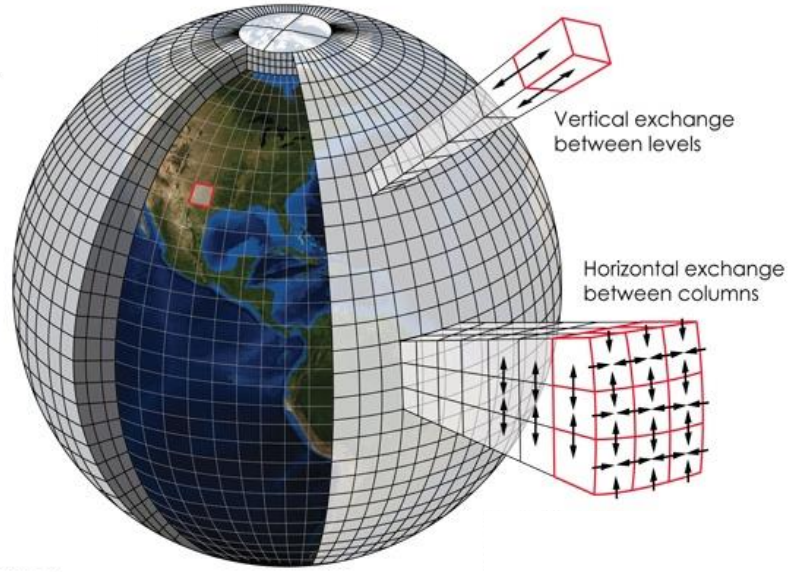
[1] **W. Deconinck et al.** "Atlas: A library for numerical weather prediction and climate modelling". *Computer Physics Communications*, 2017.

Loki – A source-to-source translation tool

Loki: Source-to-source translation package written in Python



- Library of tools and APIs to build source transformation recipes
- **Recipes depend on "Single Column" formatted code** [2]
  - Additional coding conventions that allow tools to re-optimize code for GPUs via loops flips and changing variable declarations
  - SIMD to SIMT conversion with IFS-specific memory layout (memory blocking with packed arrays and sparse slices)
- See next talk "**Cross-platform optimisation for GPUs of various flavours (the low-level tech overview)**" by A. Nawab



[2] **V. Clement et al.** "The CLAW DSL: Abstractions for Performance Portable Weather and Climate Models". *PASC*, 2018.

GPU-porting status of IFS forecast benchmark

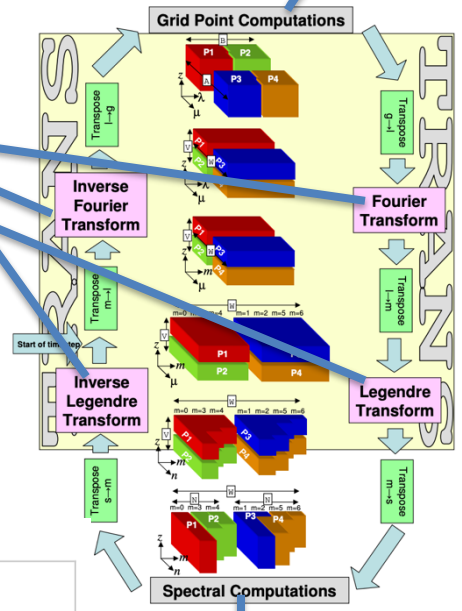
Model component		Porting method	CPU run time	GPU (Nvidia)				GPU (AMD)	
				Status	Performance	Release Cycle		Status	Perf.
Dynamical core	Spectral Transform	Manual, OpenACC	16%	Done	Good			Done	Good
	Grid point dynamics	FIELD API + Loki	10%	Done	Optimising	50r3			
	Semi-Lagrangian	Manual + Loki	12%	Done	Optimising	50r3			
	Semi-Implicit	Manual + Fxtran(MF)	2%	Integrating		50r3			
Physics	EC-physics	FIELD API + Loki	30%	Done	Optimising	49r3	50r3	Porting (AOMP)	
	Surface model	FIELD API + Loki		Done	Good	49r3	50r3		
	Radiation	Manual / Loki	5%	Done	Loki / Atlas	49r3	50r3	Integrating	
	Perturbation	Manual + Loki	N/A	Done	Good				
Wave model	Dy-core	Manual + Loki	8%	Done	Good			Integrating	AOMP-Latest
	Source term	FIELD API + Loki							
Atmospheric composition		FIELD API + Loki	N/A						
Diagnostics	DDH, FULLPOS	CPU, or manual	N/A						
Ocean model (NEMO)		~~Psyclone~~, Manual	6%	Porting (Cray, AOMP)					

Complete	Demonstrated	Working on it	External issues	Not started yet	Out of scope

Dynamical core and transforms

EcTrans: Spectral transform library

- Separate CPU and GPU code paths
- GPU optimisations include GPU-aware MPI, use of optimized math and graph libraries
- Available as standalone test and benchmark
- **Talk by L. Mosimann + poster by S. Hatfield!**

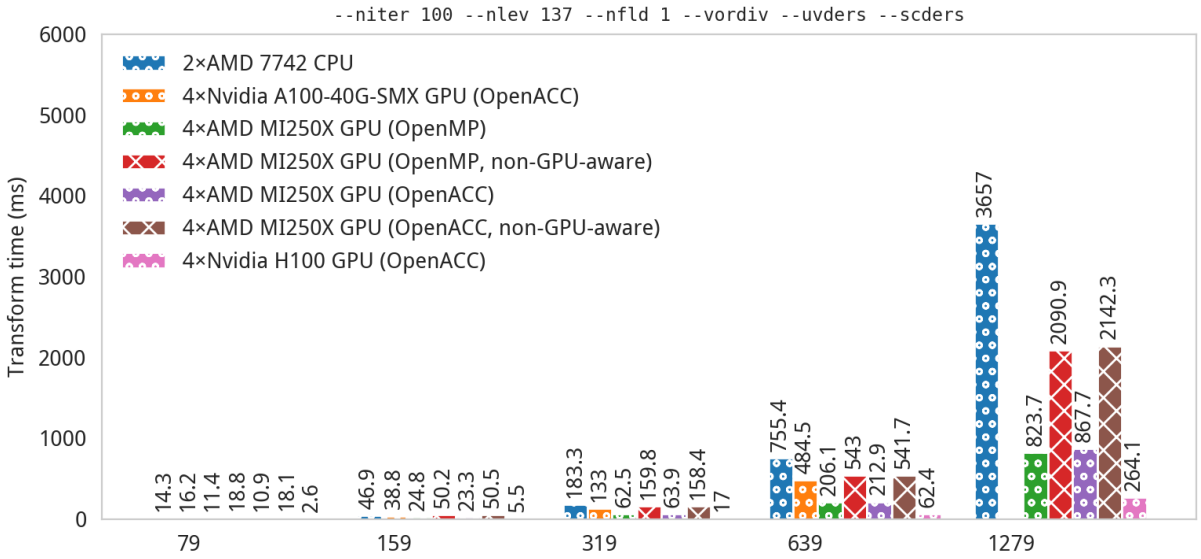


Grid-point computation (CPG)

- **Loki**: Adapted single-column recipe
- **Field-API:** Device-resident allocations

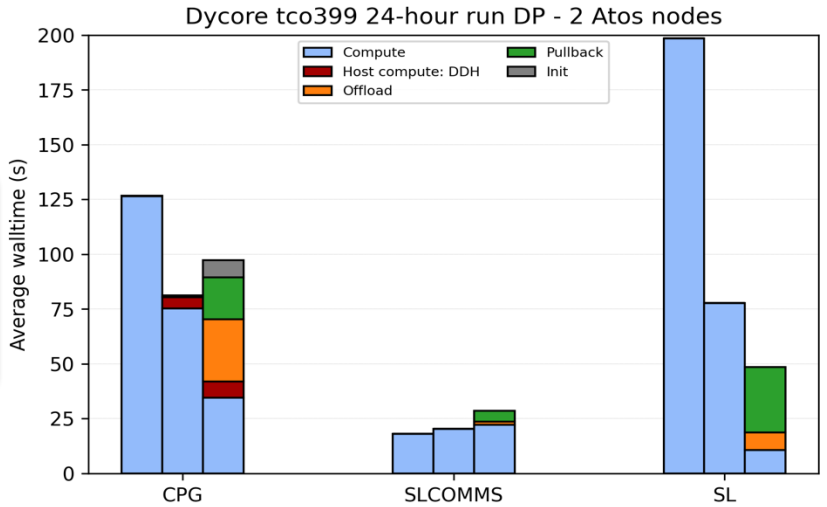
Semi-Lagrangian advection

- **Loki**: Adapted single-column recipe
- **Field-API:** Buffer allocations on device and interfacing to GPU-aware MPI



Work in progress

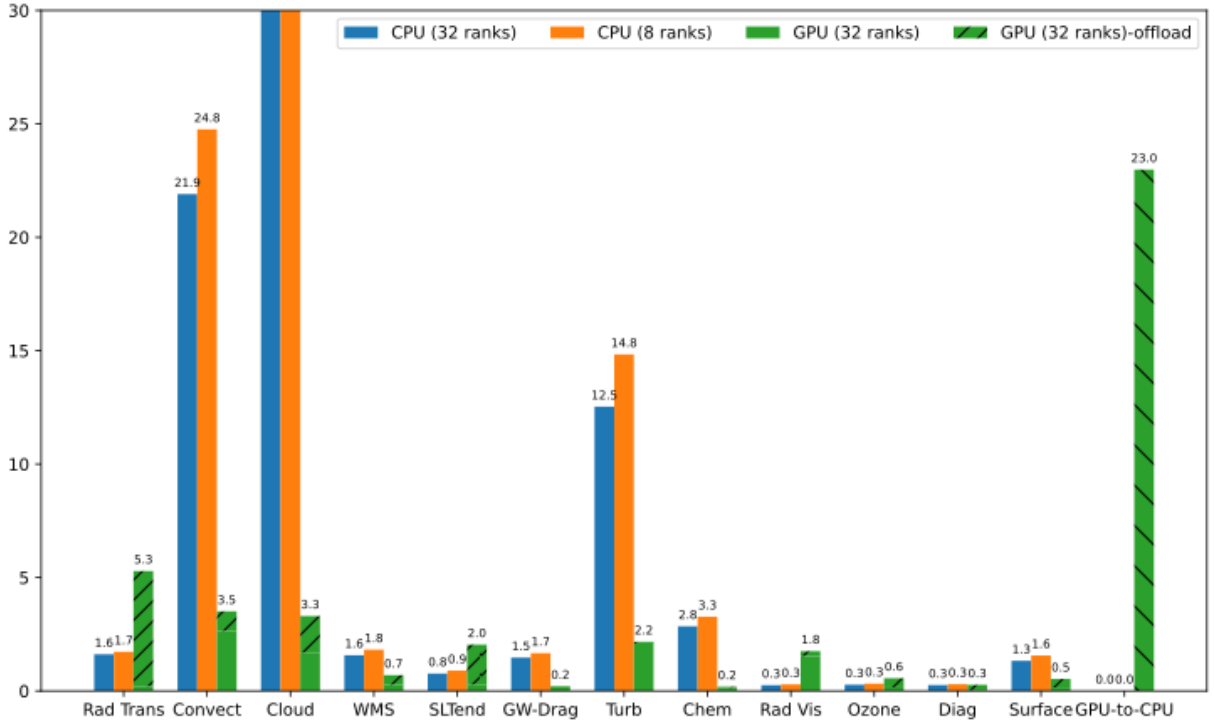
- Semi-Implicit (integration)
- Mass fixers (optional)



Left to right: CPU baseline, Loki opt. CPU, Loki opt. GPU. CPU config: 32 MPI ranks, 8 OMP threads, NPROMA 16. GPU config: 8 MPI ranks, 32 OMP threads, NPROMA 32.

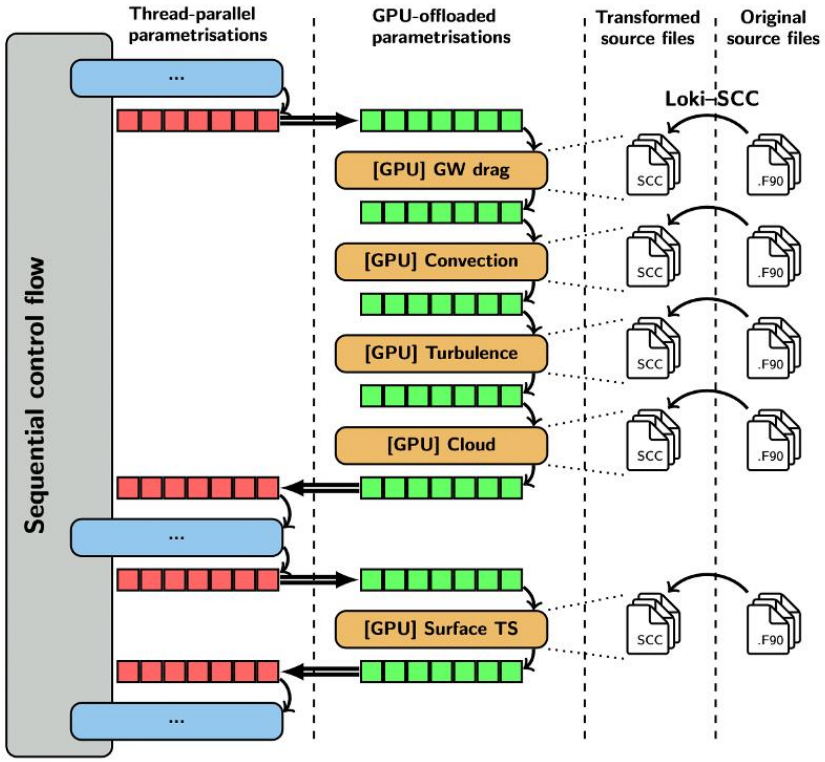
GPU-adaptation of IFS physics package

- **Auto-generated GPU code via Loki recipes**
  - Optimization passes programmed as pipelines
  - Full physics package for Cy49r1 and Cy50r1
  - More optimization and automation ongoing



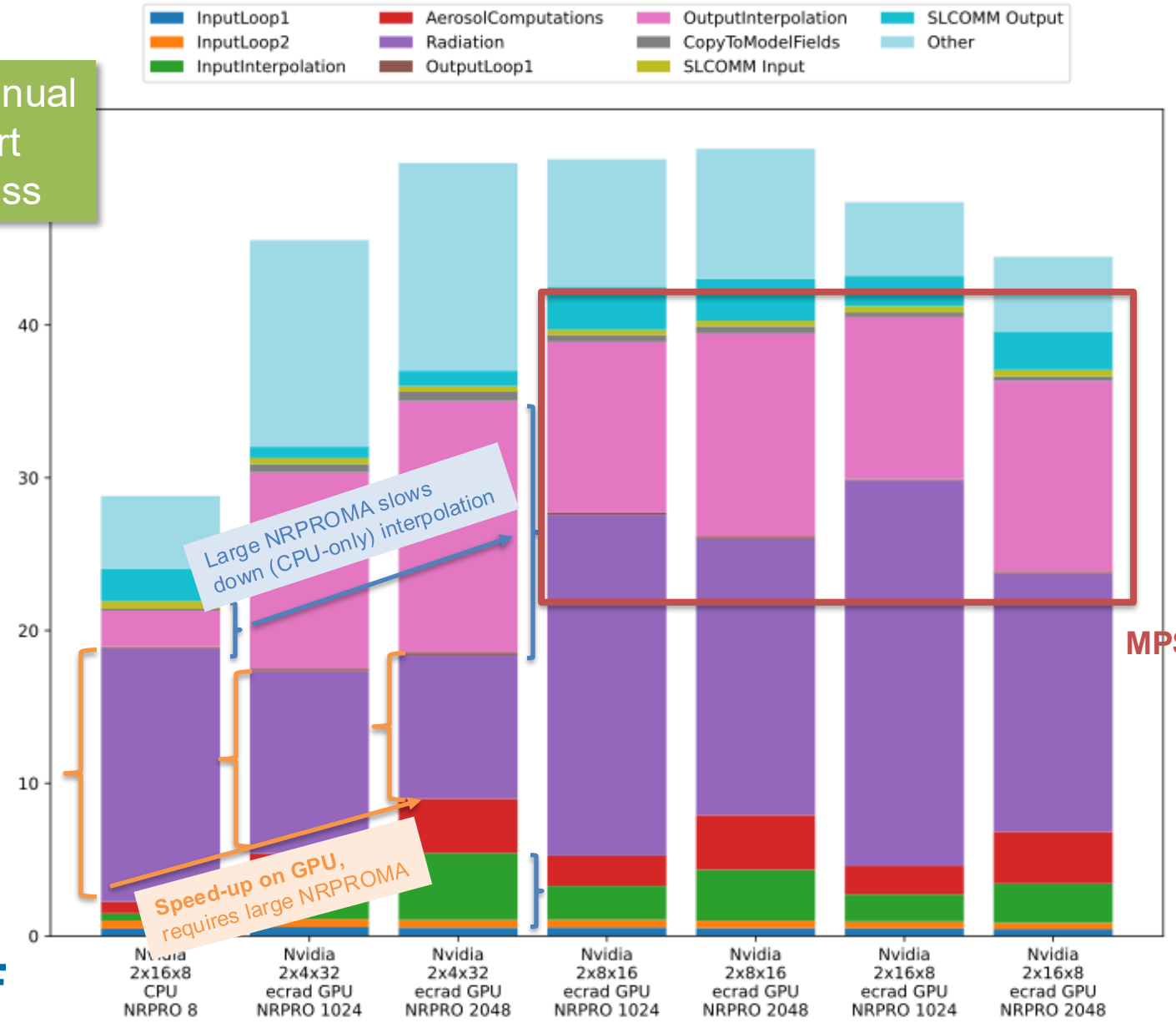
GPU development cycle

- Break single parallel loop into many
- Adapt kernels individually, verify!
- Use Loki + Field-API to offload
- Remove data transfers to GPU



ecRad - atmospheric radiation scheme

Adaptation of manual OpenACC port from MeteoSwiss



Interpolation to and from a coarser grid

Radiation scheme

MPS does not help

Ongoing work:

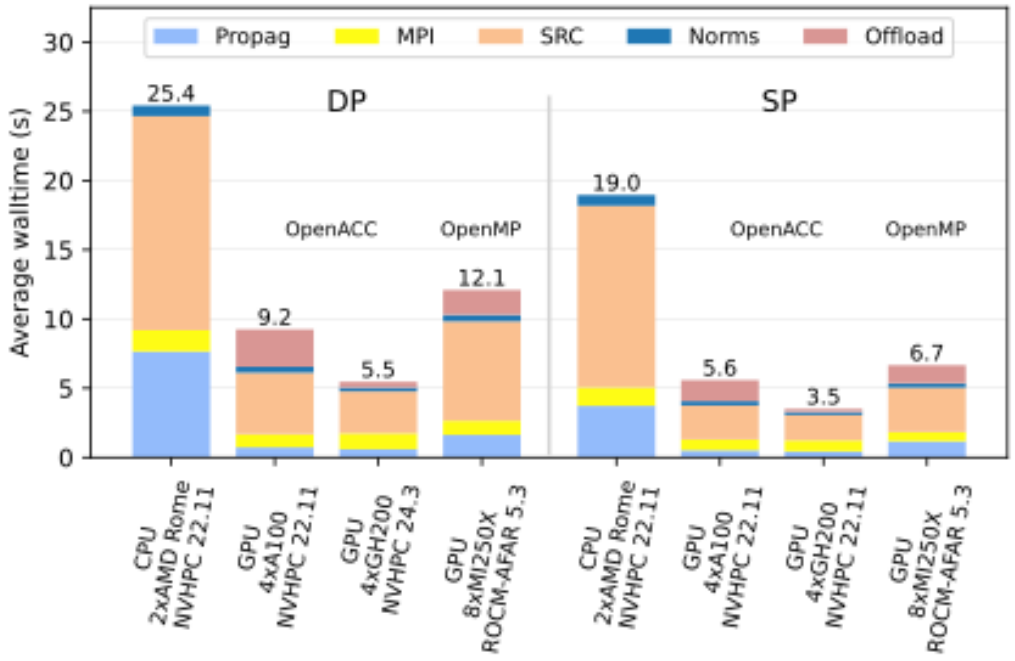
Development of a sustainable, **Loki-based GPU port** using SCC recipes for smaller NRPROMA

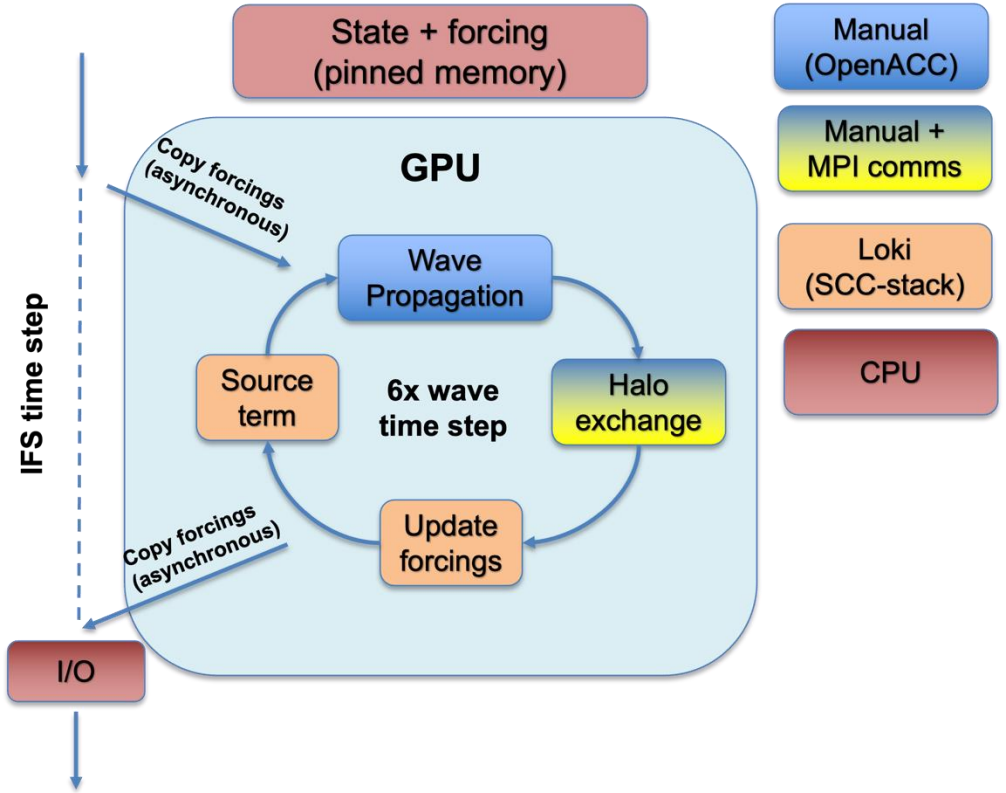
Flexible (e.g., LAM support), **GPU-capable interpolation method based on Atlas**

ECMWF

EcWAM - Open-source standalone wave model

- Released under Apache-2 license on Github
- GPU support via Field-API and Loki (OpenACC & OpenMP)
  - Optimised on Nvidia and AMD GPUs
- **See later talk by F. Di Sante and poster by A. Nawab!**





GPU-adaptation of NEMO ocean model

- **Initial collaboration with STFC on PsyClone for NEMO-5**
  - Good progress towards long-term sustainable GPU-port

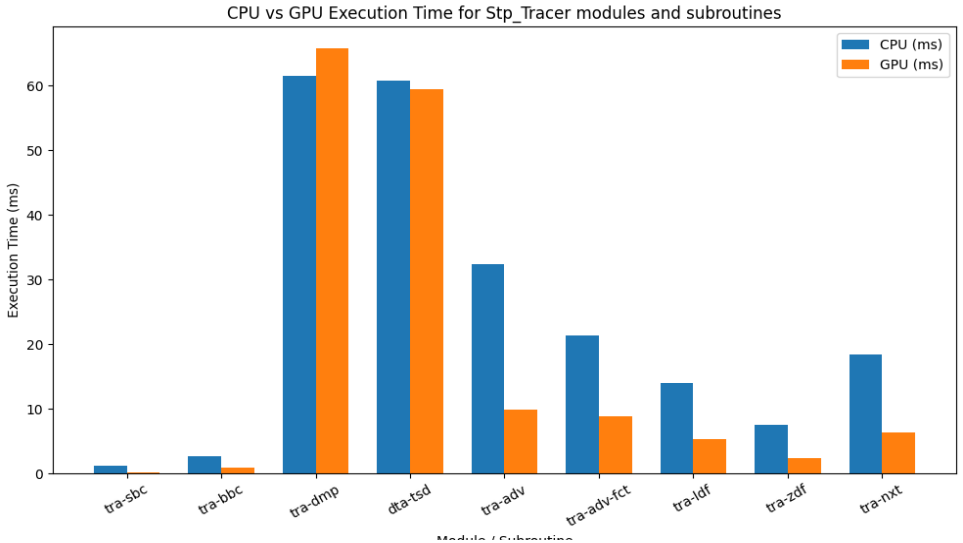
- **New focus:** Porting of NEMO-4 (OCE & LBC)
  - **Approach:** Incremental manual via OpenACC
  - **Goal:** Maximize GPU utilization

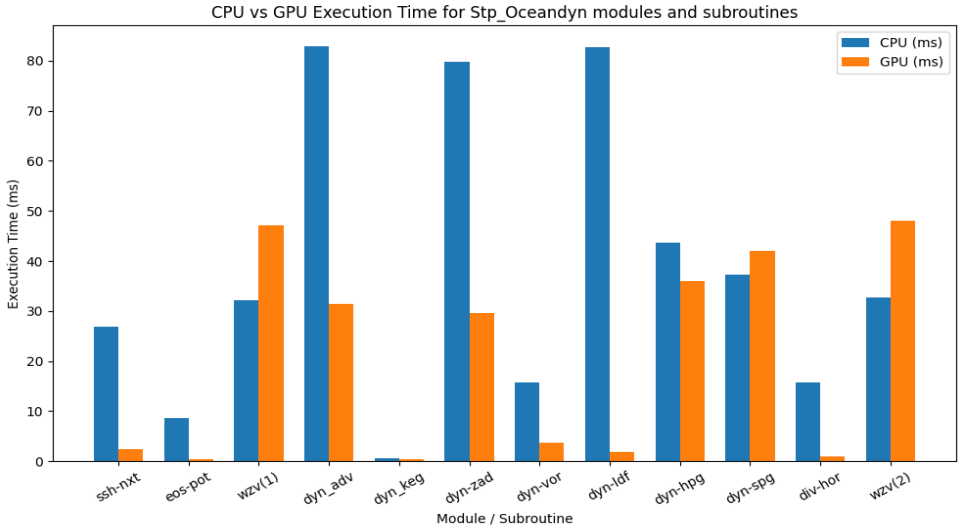
  - **Progress** on ported Modules (**ORCA1**):
    - Ocean diagnostics (**DIA**), tracer (**TRA**), diffusion (**LDF**), forcing & data (**DOM**), dynamics (**DYN**)

  - **In Progress: ZDF** (Vertical Diffusion) modules

- **Up next:** Testing higher resolution case: **ORCA025**
- **Up next:** Performance profiling & scalability tests

- **Open question:** Long-term sustainable multi-arch version?



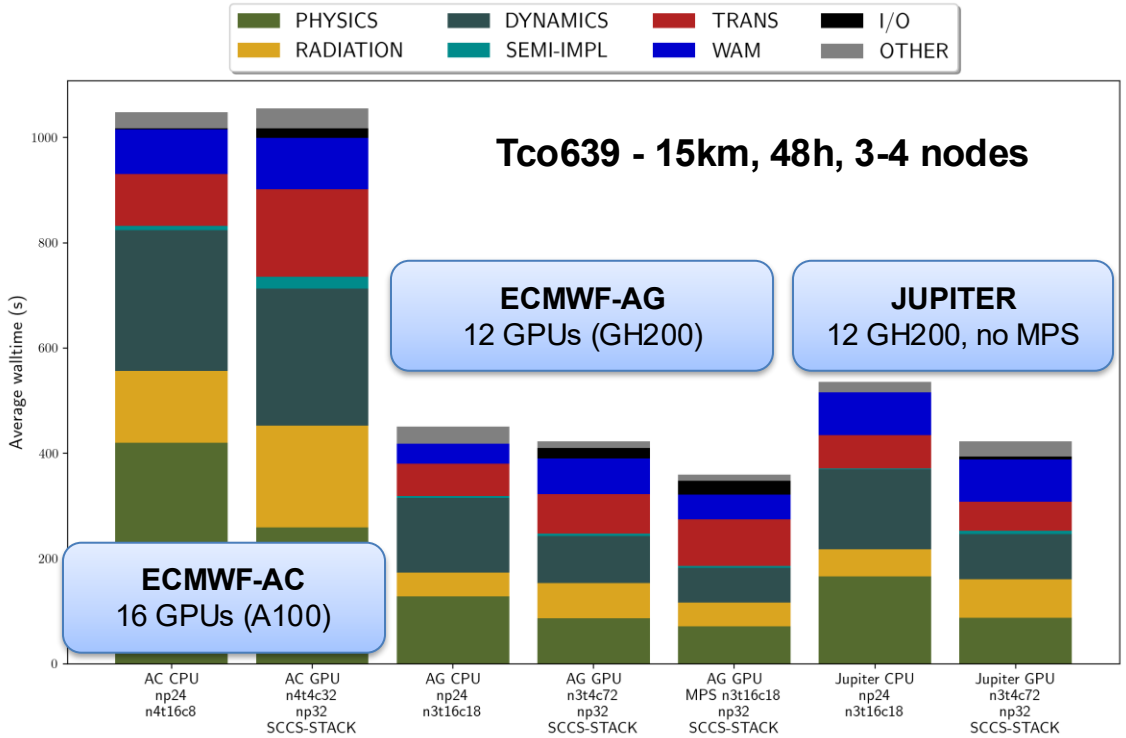


GPU-adapted IFS benchmark - putting it all together

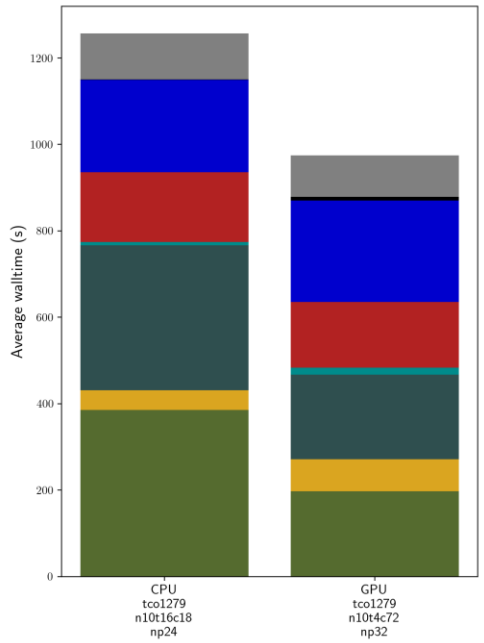
Latest RAPS release – internal cycle Cy50r3

- High device residence (>85%), optimization ongoing
- Initial runs competitive despite wasteful data transfers!
  - GPU over-subscription via MPS very beneficial!

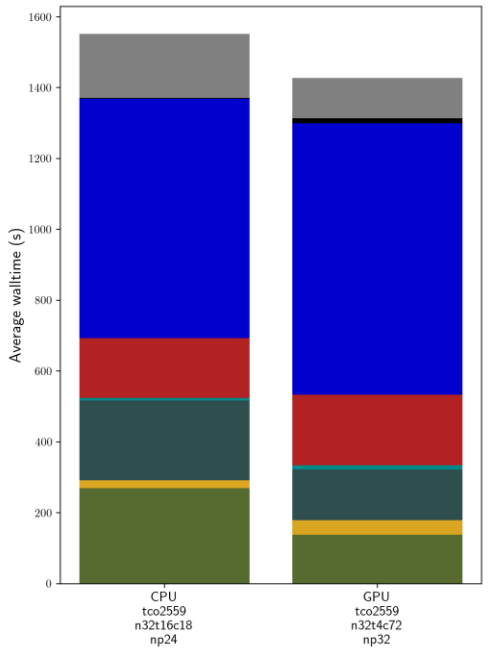
GPU-enabled IFS
- Competitive on Grace-Hopper
- Under-optimised data transfers
- Derived from optimised CPU
- **Latest science cycle (Cy50r1)**



Tco1279 - 9km, 48h
10 nodes of GH200



Tco2559 – 4.5km, 12h
32 nodes of GH200



GPU-adapted IFS benchmark – optimization ongoing

More throughput gains expected

Redundant data transfers

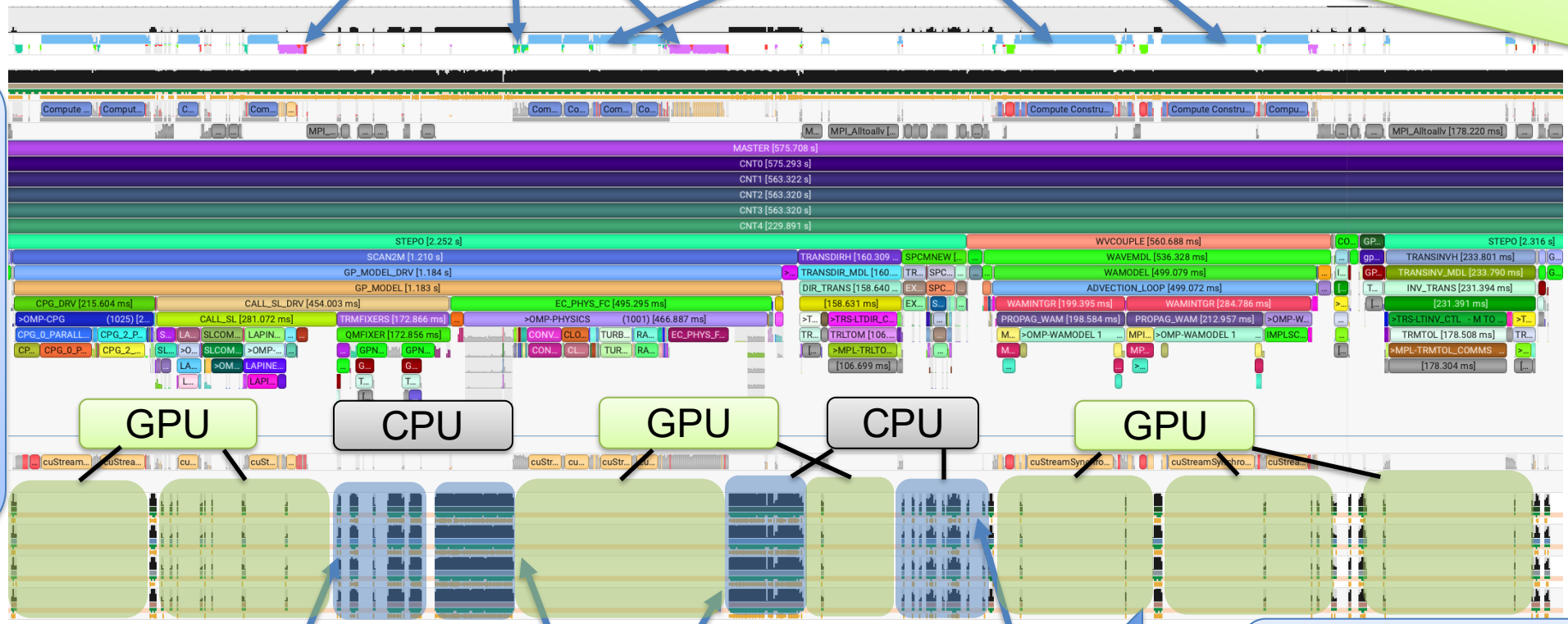
High device-residency; fast compute but more improvements coming

Optimisation performed on latest science, adjacent to operational CPU code!

Atmospheric timestep

9k resolution
10 nodes
GH200
JUPITER

Incl. wave
No MPS



Optional mass fixers

Bottlenecks in physics-dynamics interface

Missing semi-implicit

Remaining utility loops become costly without over-subscription (MPS)

Progress on GPU adaptation of IFS

Sustainable GPU adaptation alongside scientific development

- Close collaboration with member state and strong synergies with Destination Earth
- Enable continuous adaption and performance optimization for HPC architectures

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

Progress: Most components offloaded and optimization ongoing

- GPU offload for >85% of atmospheric time step, based on latest science (Cy50r1)
- Optimisation ongoing on JUPITER and in-house GPU partitions

Thank you! Any questions?

michael.lange@ecmwf.int
MLange805