

WGNE – HPC/Exascale update

Nils Wedi

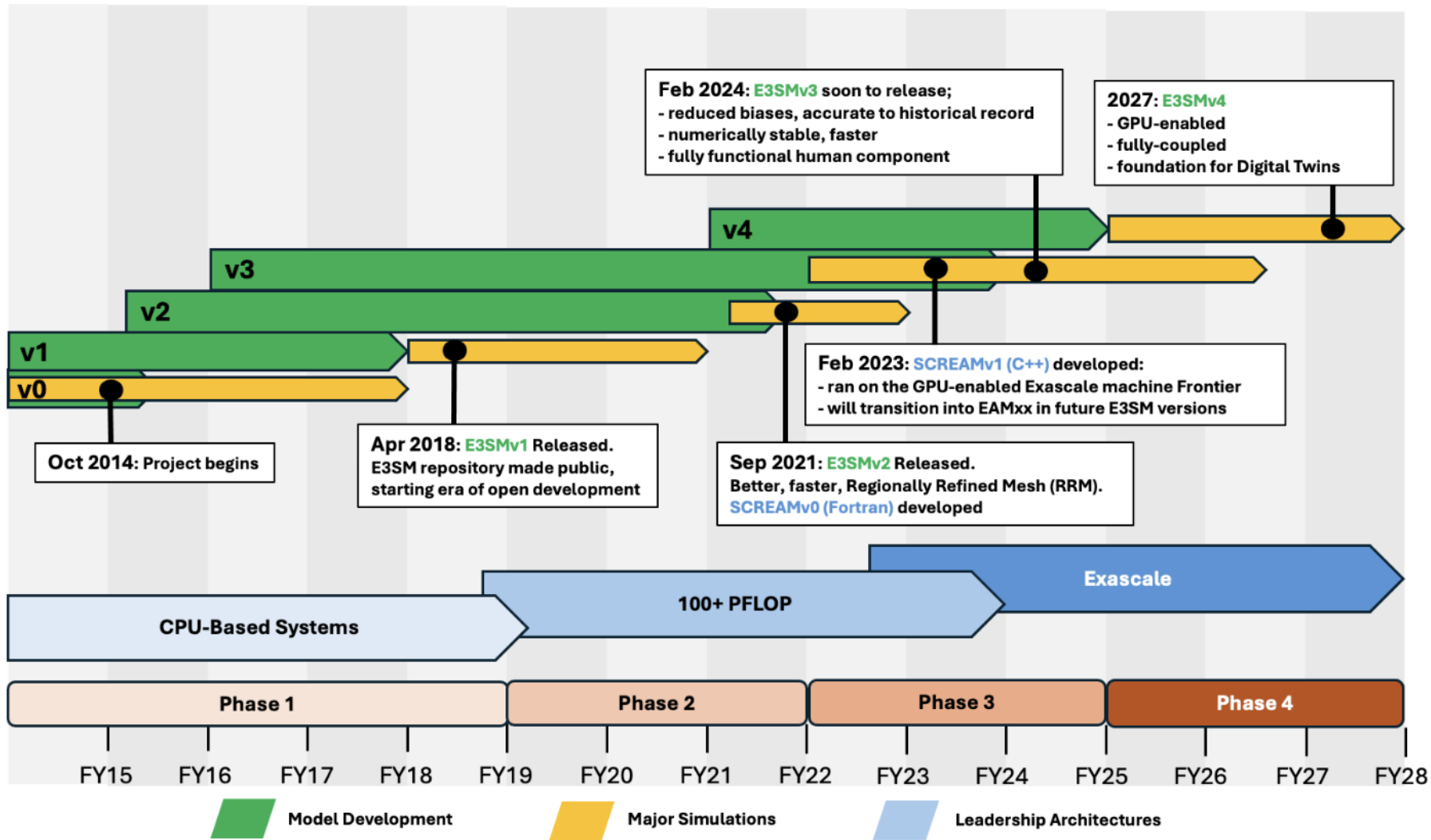
European Centre for Medium Range Weather Forecasts (ECMWF)

Many thanks for all the individual contributions!

Contents

1. Overview of trends in HPC / weather & climate preparation for Exascale
2. GPU adaptation, single precision, I/O, workflow and distributed compute
3. Maintainability / Performance portability
4. Annex: provided slides from members & other groups

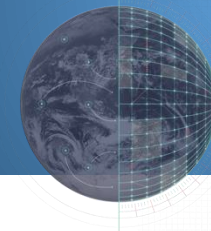
E3SM Timeline



10 years of E3SM development

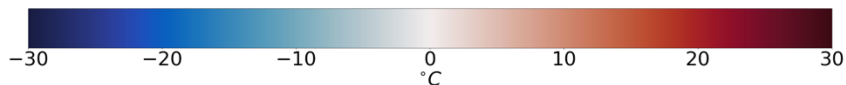
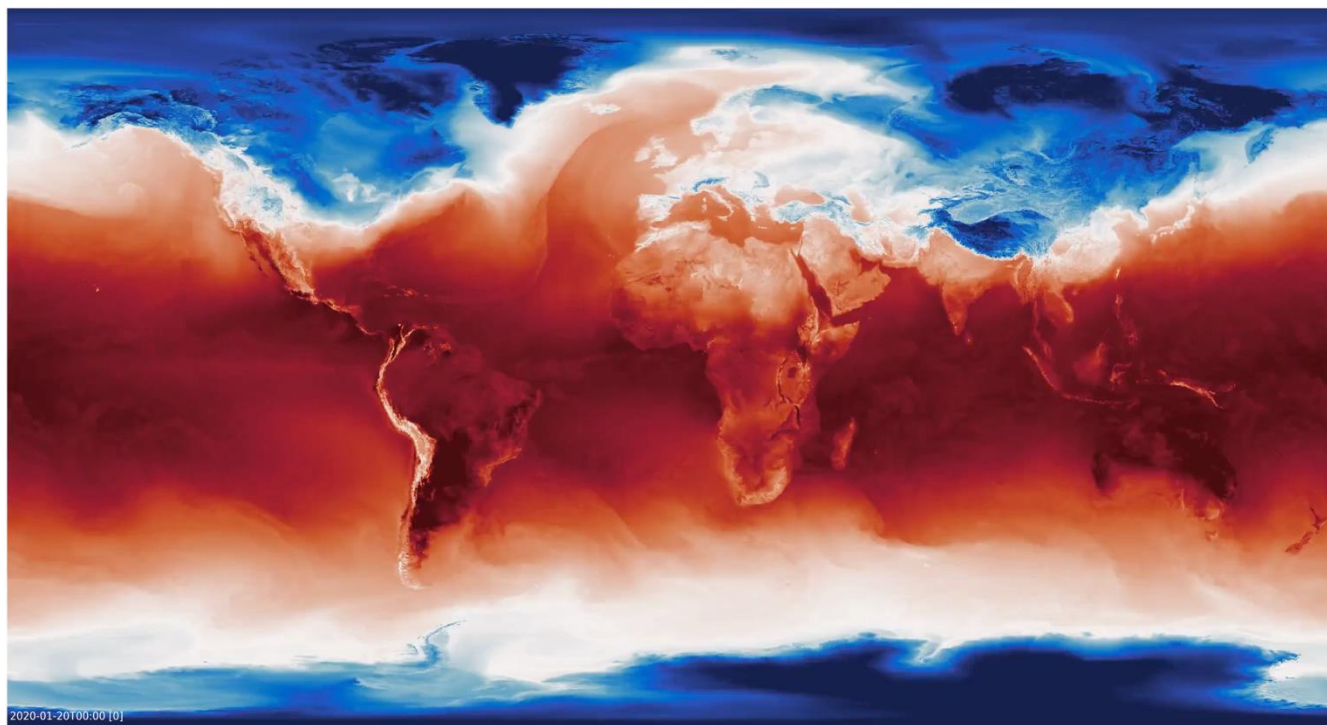
<https://e3sm.org/e3sm-a-decade-of-progress-a-timeline>





Climate Digital Twin: 1st operational capability for climate projections

To test the impact of certain events, scenarios or policy decisions on multi-decadal timescales



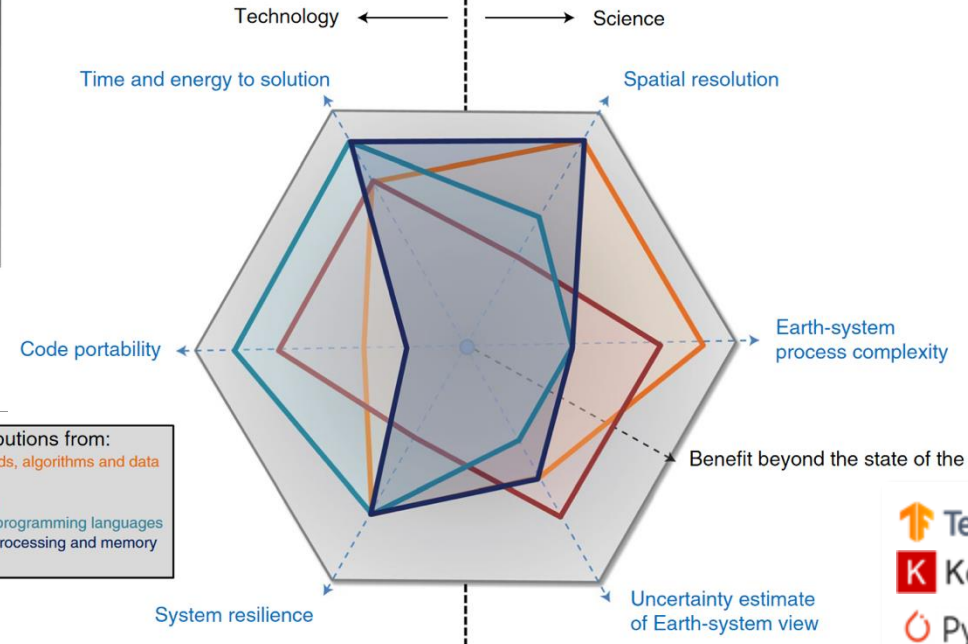
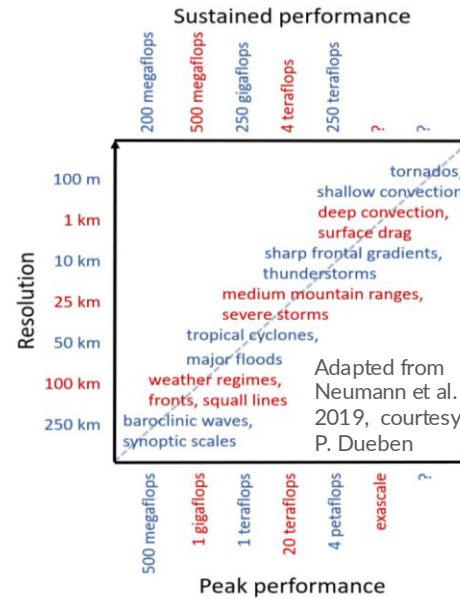
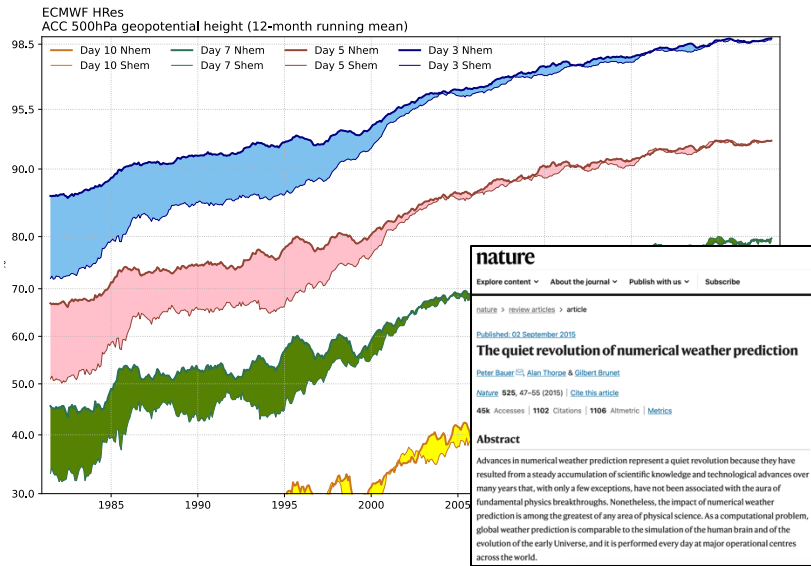
Regularly updated (yearly) and on-demand

1 x scenario 30 years

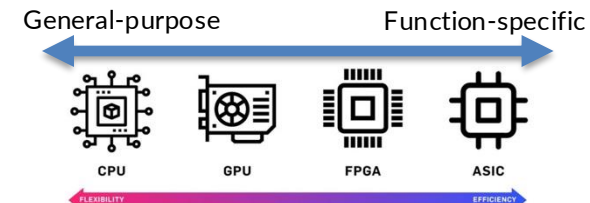
== 1-1.5 Million GPU h or 34 Million CPU-core h

IFS-NEMO or IFS-FESOM and
ICON-ICON coupled

After decades of steady progress, we are seeing challenges and opportunities



- Individual contributions from:
- Numerical methods, algorithms and data structures
 - Machine learning
 - Domain-specific programming languages
 - Heterogeneous processing and memory architectures



Source: venturebeat.com

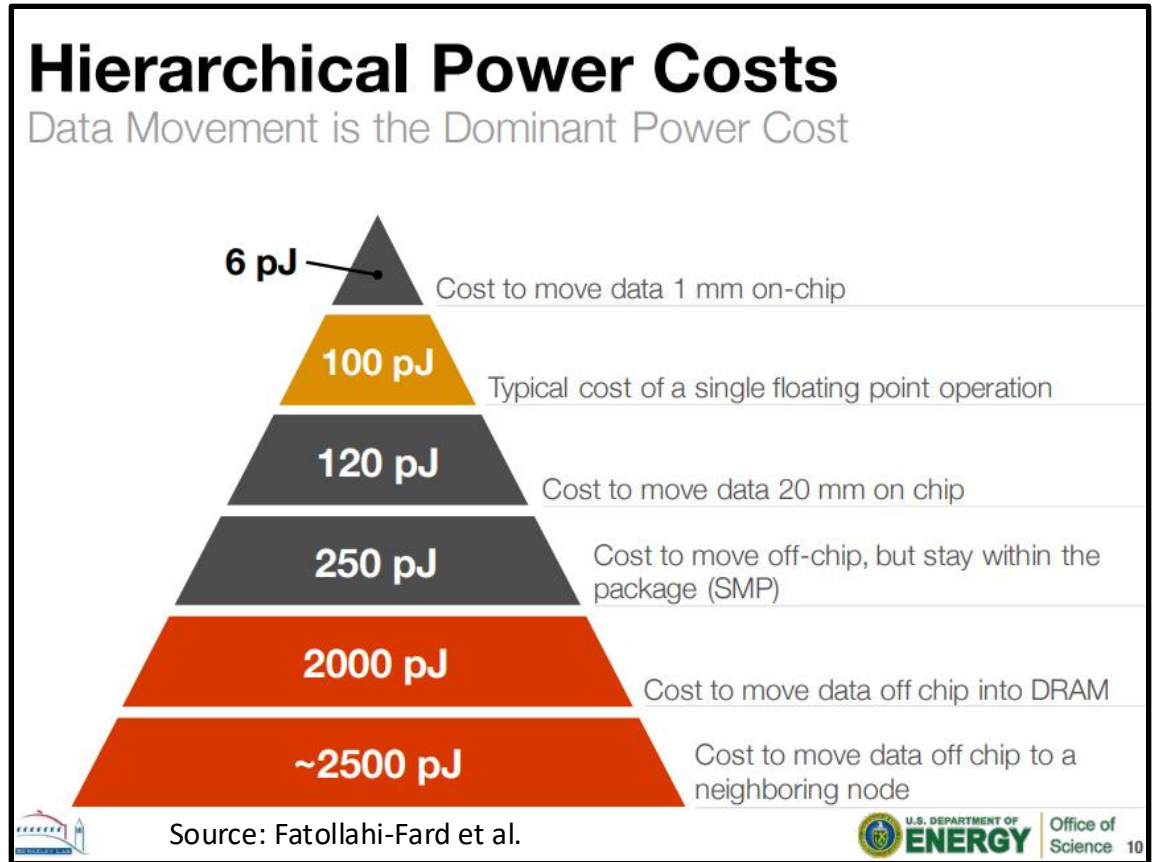
- ❖ Novel technologies offer great potential for higher physical fidelity and energy efficiency. At the same time, atmospheric models and ESMs face an increasingly diverse landscape of supercomputing architectures.
- ❖ Efficient execution requires targeted hardware-specific implementation and optimization. Serving various hardware and computing systems involves highly complex code that needs to be organized well to maintain productivity.
- ❖ Emerging and future hardware is heterogeneous with specialized and energy-efficient parallel entities
- ❖ AI / Machine Learning will have a significant impact on hardware potentially towards high flop-rate at low precision, optimization of data movement becoming even more important

Energy cost of simulations

Alexandru Calotoiu, Thorsten Hoefler et al
Presented at PASC 2024

Stateful Dataflow Multigraphs: A Data-Centric Model for Performance Portability on Heterogeneous Architectures

[Tal Ben-Nun](#), [Johannes de Fine Licht](#), [Alexandros Nikolaos Ziogas](#), [Timo Schneider](#), [Torsten Hoefler](#)

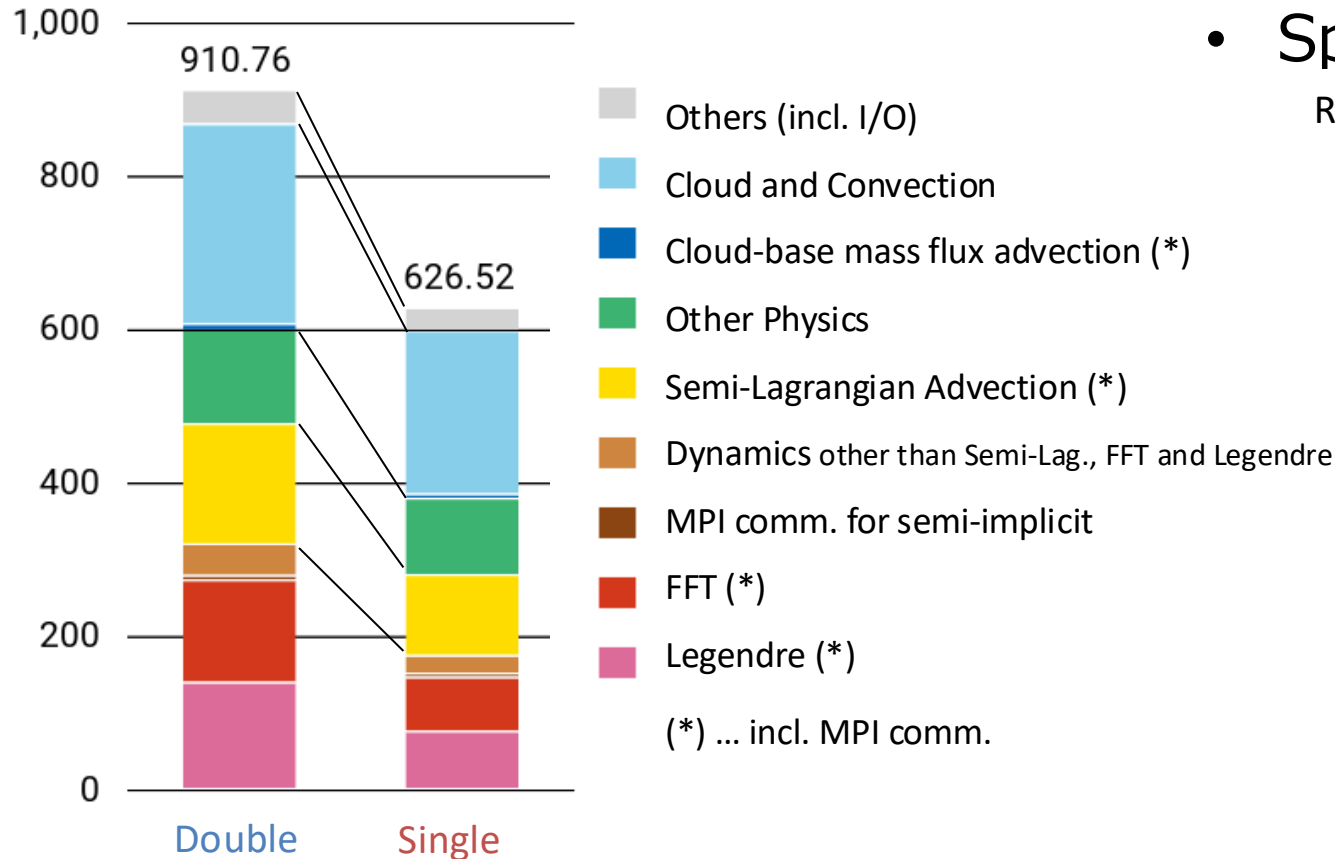


Understanding data movement is key to performance & portability

e.g. DaCe: <https://github.com/spcl/dace>

Single precision GSM

See also detailed slides for Tips & Tricks !!



MPI rank average elapsed time [s] of Tq959 GSM (dx~13km) for 132hr time integration

49nodes, 390ranks (incl. 6 I/O ranks) and 14 OpenMP threads on Fujitsu PRIMAGY CX2550 M7, Intel Fortran(2021.9.0) with "-O2" option

- Speed-up by **30%**

Ratio of elapsed time in single precision GSM to double precision

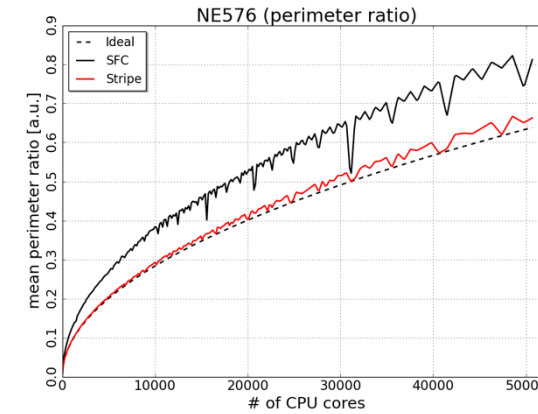
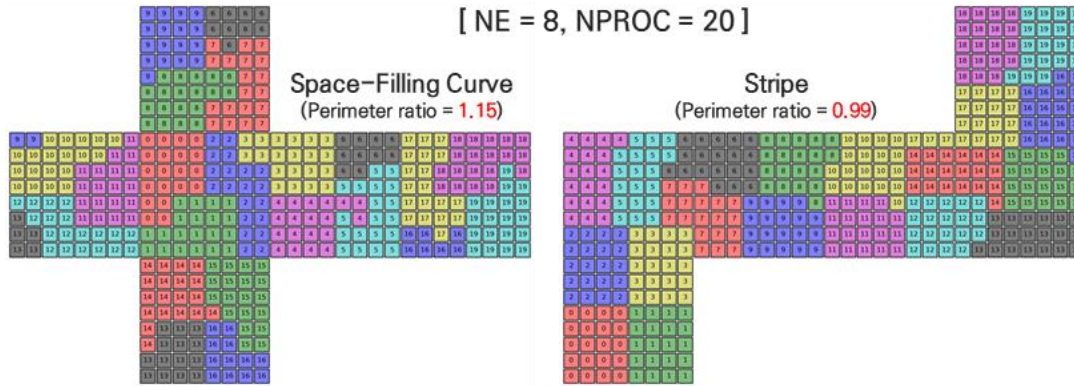
Process	Single/Double
Others (incl. I/O)	0.67
Cloud and Convection	0.82
Other physics	0.81
Semi-Lagrangian Advection	0.67
Other dynamics	0.54

Small speed-up rates in physics parameterization, particularly in specific subroutines presumably due to:

- Slow convergence in iterative algorithms
- SIMD suppression in loops with complex "if" branches

New grid partitioning “Stripe method”

: significantly reduces the model's communication amount (communication \propto perimeter rate)



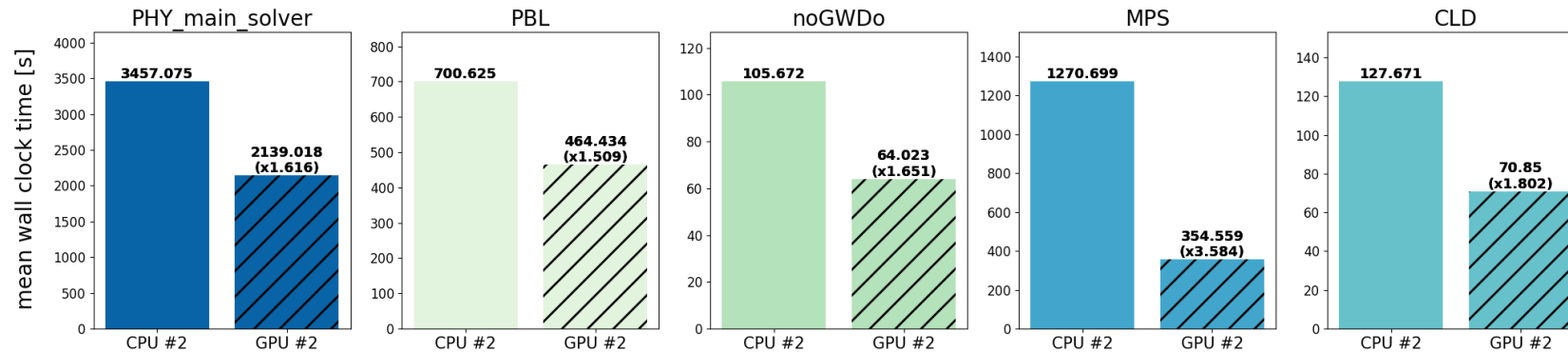
[Space-Filling Curve (old) vs Strip method (new) in grid partitioning]

- KIM: Korean Integrated Model developed by KIAPS using Cubed-sphere grid

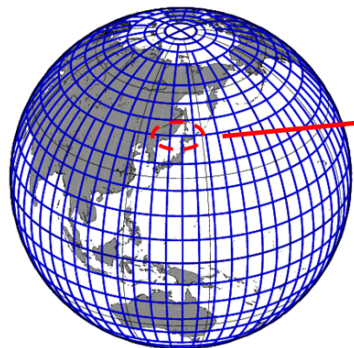
GPU porting for KIM physics

: achieved about 1.6x speed-up when using the GPU

CPU: Intel Xeon Broadwell E5-2620v4 x 2, GPU: NVIDIA V100 x 2

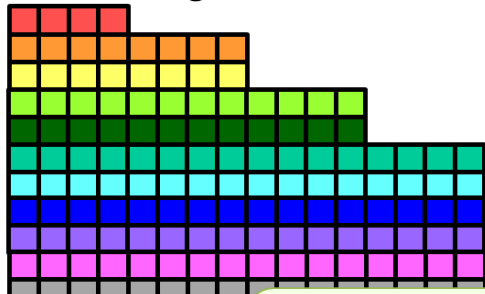


Tq959L128 960mpi



Lon. →
Lat. ↓

Horizontal grid boxes in a MPI process



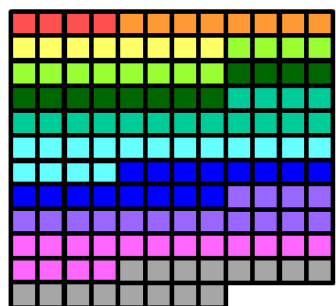
NUMI_I=12

NUMI_I=36

Max array size of array in the "i" direction is controlled by a parameter "NUMI_I"

For CPU with OpenMP
(larger outermost loops
for thread parallelization)

For Vector machines or GPU with OpenACC:
(larger innermost loops for vectorization)



i →
j ↓

NUMI_I

NUMJ_S

Typical source code in GSM

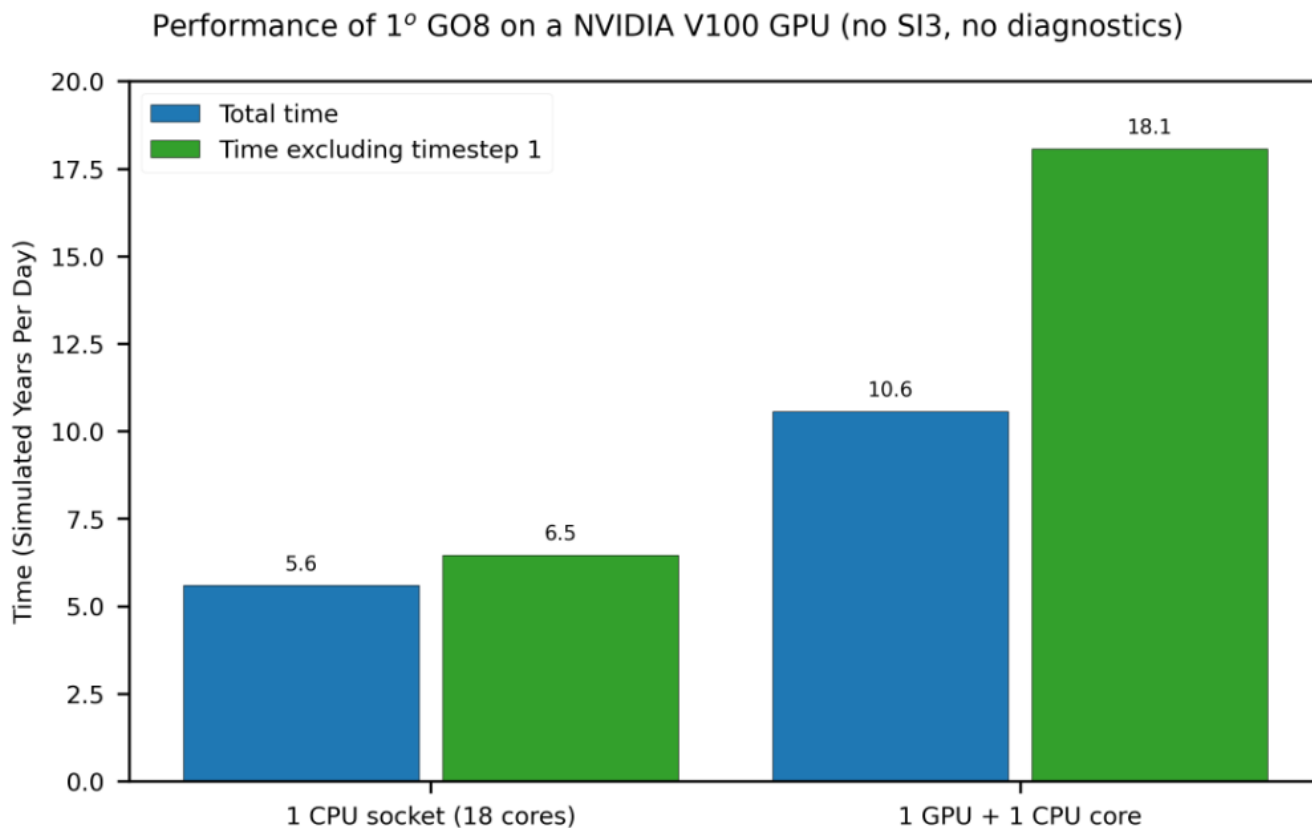
```
integer(4) :: WPR = kind(1.0d0) ! double
real(kind=WPR), dimension(NUMI_I, NUMFA_I, NUMJ_S) ::
array1(:,:,:),array2(:,:,:)
!$OMP PARALLEL default(SHARED), private(i,k,j)
!$OMP DO schedule(DYNAMIC)
do j = 1, NUMJ_S
  !$acc kernels &
  !$acc present(NUMI,array1, array2,...)
  do k = 1, NUMFA_I
    do i = 1, NUMI(j)
      array1(i,k,j) = "parallel calculation using array2(l,k,j)"
    end do
  end do
  !$acc end kernels
end do
!$OMP END DO
!$OMP END PARALLEL
```

Elapsed time [s] of Tl159L128 GSM (~110km) for 6hour time integration (1node, 8MPI, 14threads, 8GPU)

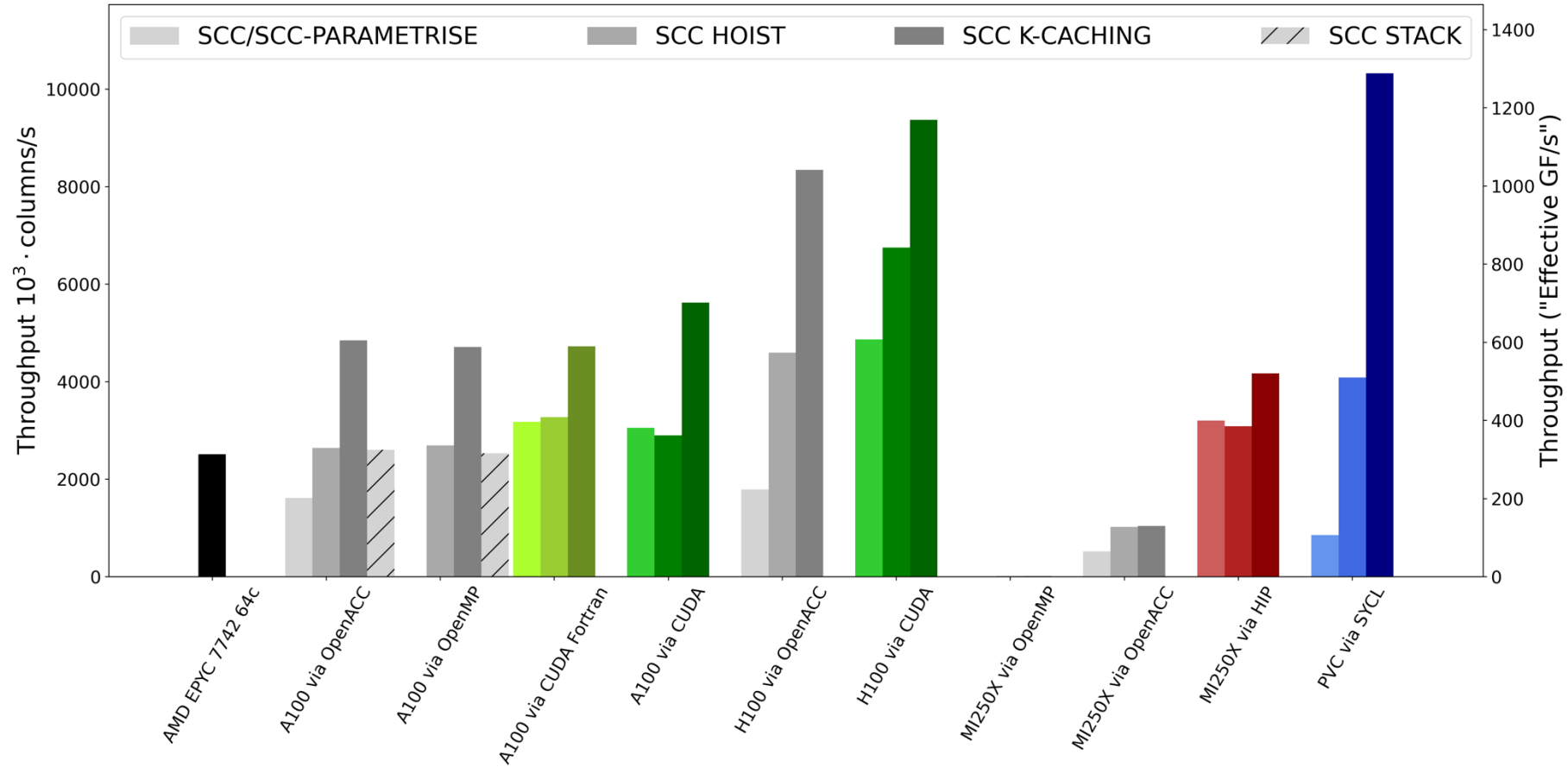
	All CPU (optimized outermost loop for CPU(OpenMP))	Semi-Lag. on GPU, others on CPU (optimized innermost loop for GPU(OpenACC))
Semi-Lagrangian advection	2.9413	1.5076 😊
Cloud and convection	1.0733	13.881 😞
Other physics	2.3951	20.433 😞

Optimized for GPU, but awful performance in CPU

NEMO performance on GPUs



- [PSyclone](#) is used by NEMO and several other models to transform their codes for running on GPUs
- STFC's work for the [ExCALIBUR project](#) demonstrated that an ocean-only simulation based on NEMO 4.0.2:
 - Runs ~3.4x faster on a NVIDIA V100 GPU than on a 16-core Intel Skylake CPU (1° grid)
 - Runs on 90 NVIDIA V100 GPUs with performance equivalent to 270 16-core Intel Skylake CPUs (1/12° grid)
- Recent work under the [Met Office NGMS programme](#) achieved a similar (~2.8x) performance increase for NEMO 4.0.4
 - 1 NVIDIA V100 GPU vs 1 18-core Intel Cascade Lake CPU
- This is currently being repeated for NEMO 5.0



So many variants – why can't compilers do this for us?

[1] M. Staneker et al. "CLOUDSC cross-platform performance study". In preparation for GMD.

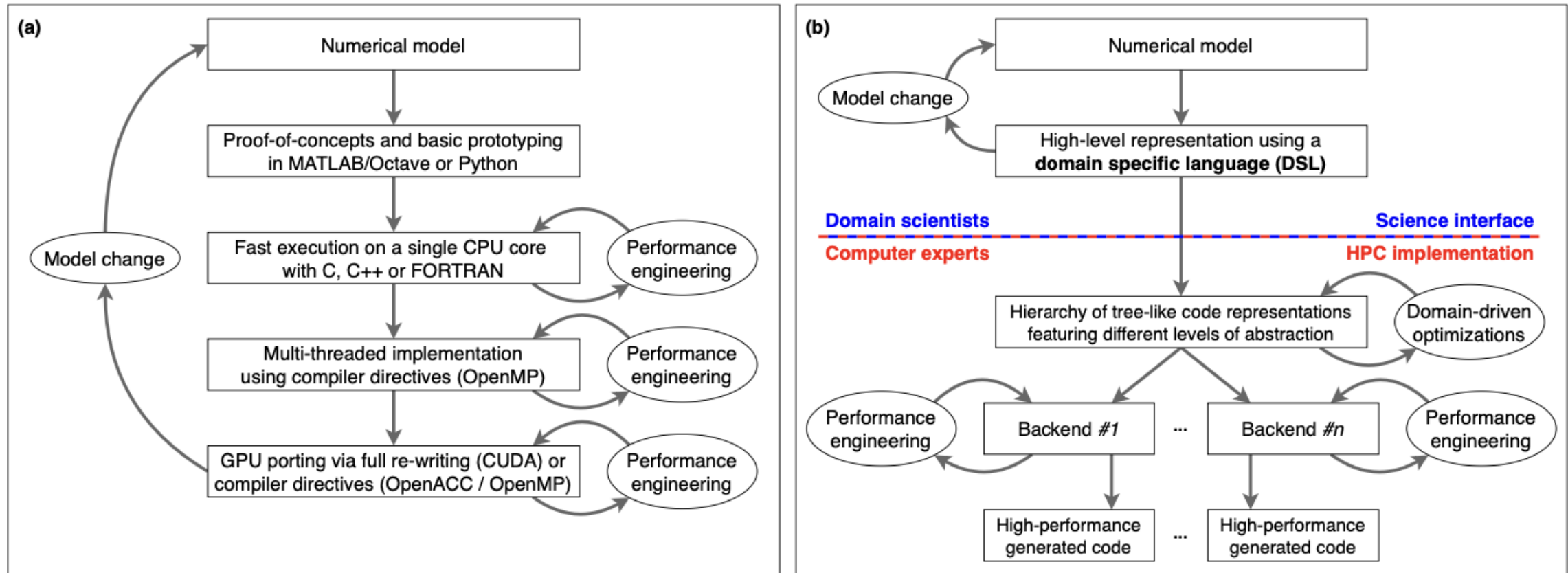
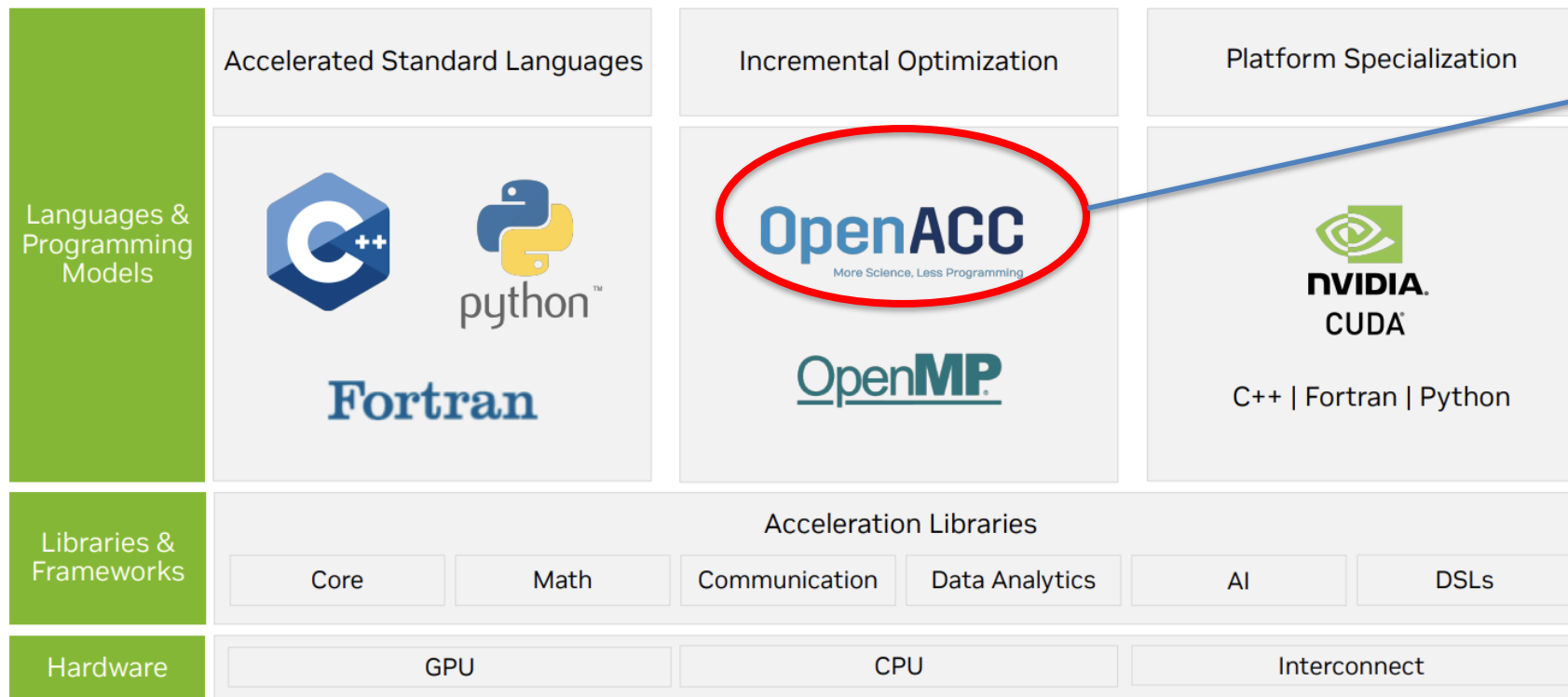


Figure 1. Diagrams comparing (a) a well-established workflow in scientific software development, and (b) a DSL-based approach resembling the software engineering strategy advocated in this paper. The red-and-blue dashed line in (b) mark the separation-of-concerns between the domain scientists and the computer experts.

Programming the NVIDIA Platform

Unmatched developer flexibility



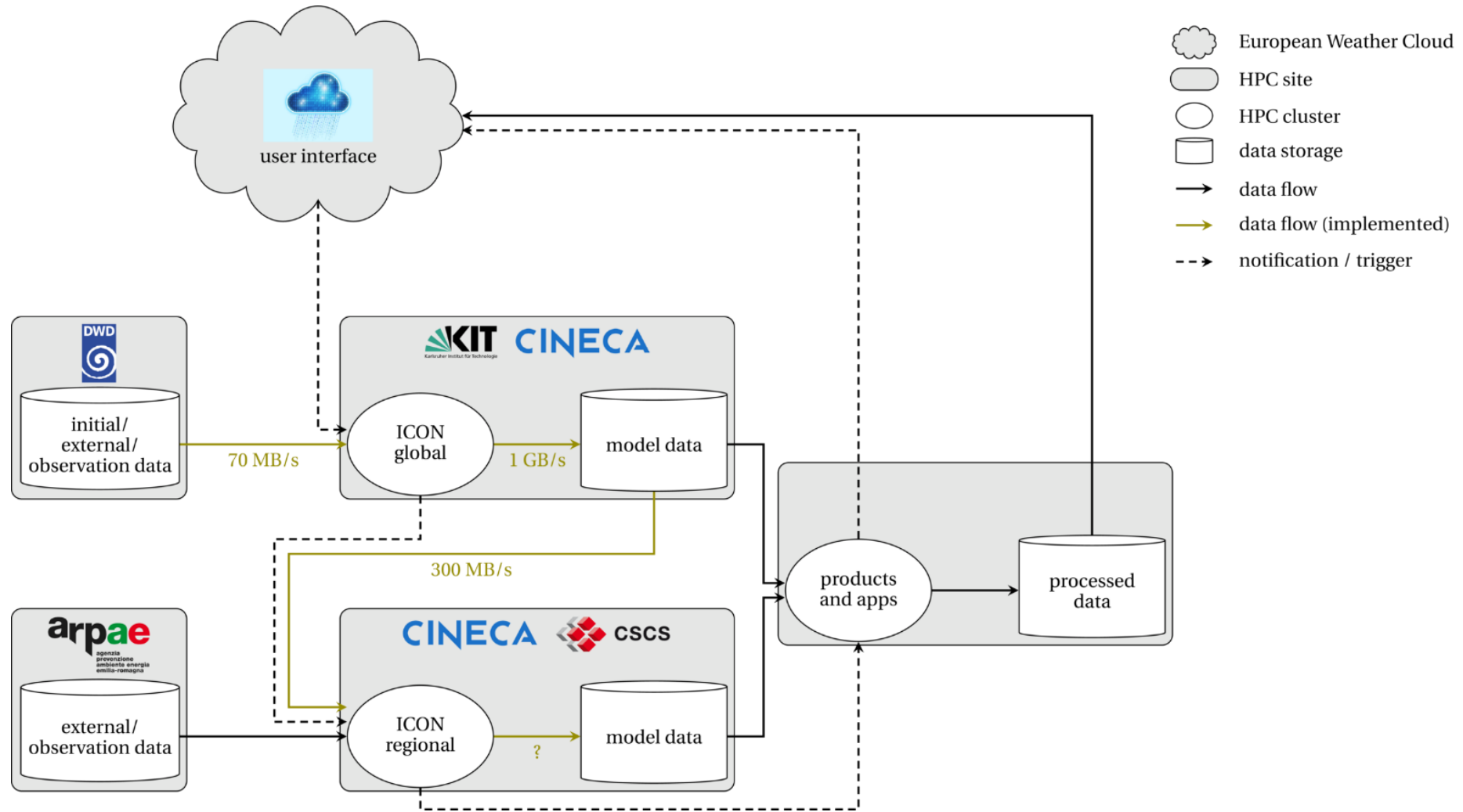
“Turned out to be the less divergent approach acrossd EuroHPC platforms” in DestinE used by both IFS and ICON!

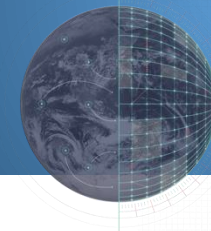
Developer tales: “AMD noticed that the Cray compiler inserts hipEventRecord/hipStreamWaitEvent calls around OpenACC kernels. The calls do not seem to serve any clear purpose, but they add latency to kernel launches. Following his suggestion, we now use a LD_PRELOADED library to replace the calls with dummy versions. This has resulted in a 10 - 15% increase in performance”

Anastasia Stulova, Nvidia, PASC2024

- Check compilers ...
- Compilers are good at optimising common structures
- Code refactoring accounts for a large part of speed-up
- Reuse, memory and data movement management not typically done by compiler

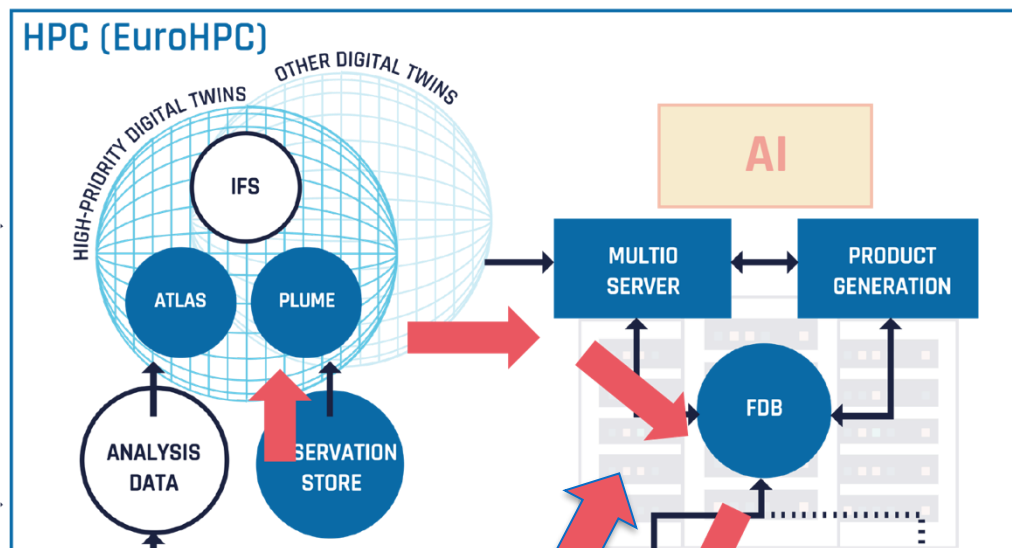
Data flow within GLORI-DT





DESTINE WORKFLOWS RESEMBLE AI - FACTORIES

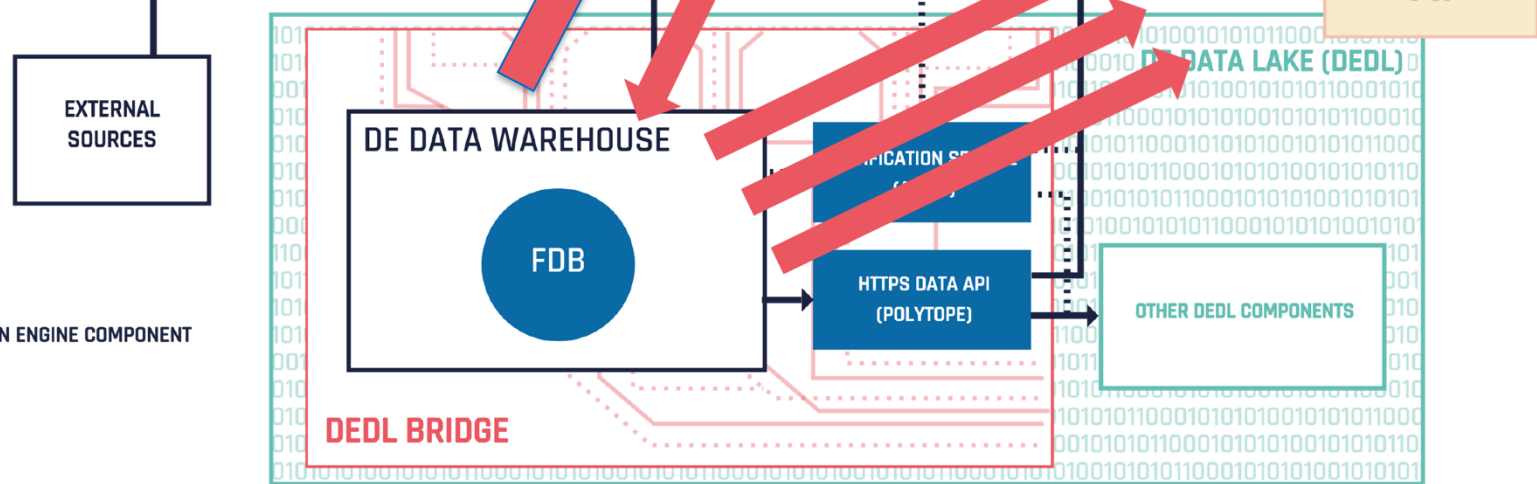
Python frameworks are used across accelerated physical modelling, ML/AI frameworks, geographically distributed (cloud) data access, processing and visualisation





WMO WIS2.0 compatible data access

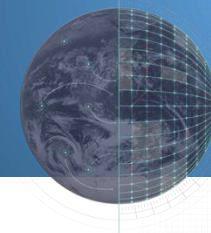
<https://pygeoapi.io/>

<https://polytope-client.readthedocs.io/en/latest>



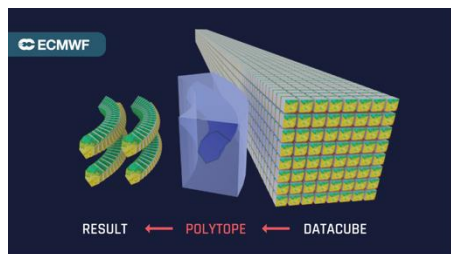
 DIGITAL TWIN ENGINE COMPONENT

 DATA FLOW

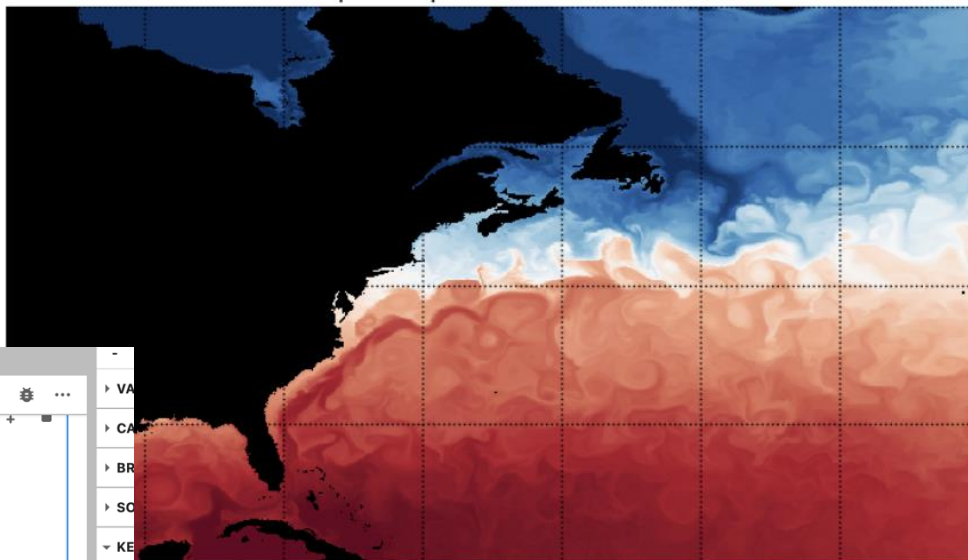


INTEROPERABLE SEMANTIC DATA ACCESS

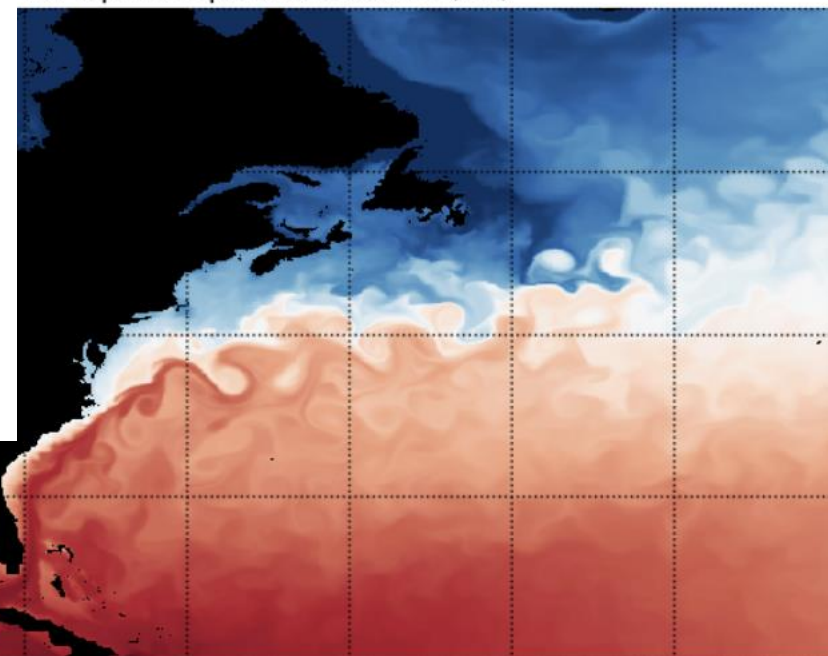
Swagger API; STAC catalogue; OGC/FAIR/INSPIRE compliant interfaces, AI-ready datasets, "truly harmonized" data access, hierarchical access, etc...



lev 2 pot-temperature over GS (ICON)



lev 2 pot-temperature over GS (IFS)



```
Terminal 1 | Healpix_ocean_example.ipynb
[3]: import earthkit

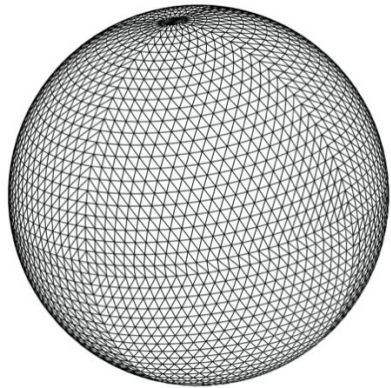
request = {
    "class": "d1",
    "dataset": "climate-dt",
    "activity": "scenariomip",
    "experiment": "ssp3-7.0",
    "realization": "1",
    "generation": "1",
    "model": "icon",
    "resolution": "high",
    "expver": "0001",
    "stream": "clte",
    "date": "20251129",
    "time": "0000",
    "type": "fc",
    "levelist": "2",
    "levtype": "o3d",
    "param": "263501"
}

data = earthkit.data.from_source("polytope", "destination-
```

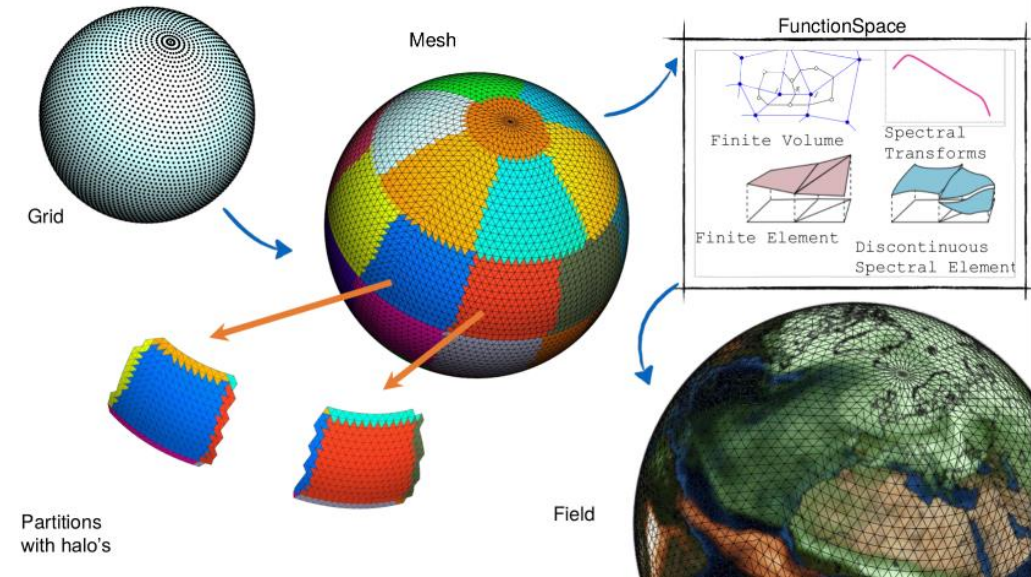
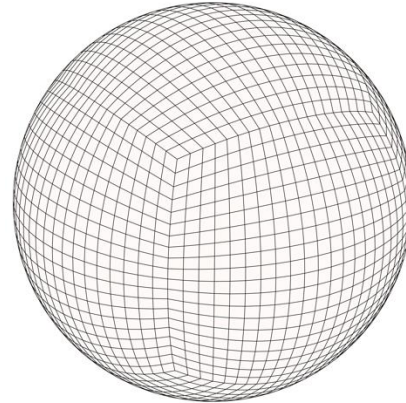
Polytope - request

Enabling new global grids and numerical schemes for (sub)-km-scale

Octahedral grid



Cubed-sphere grid

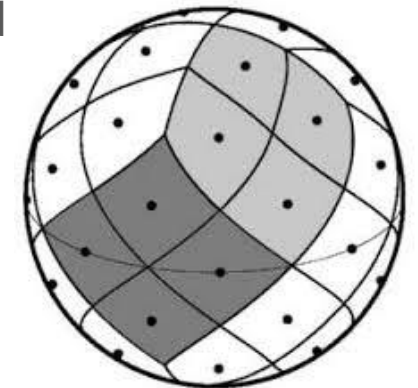
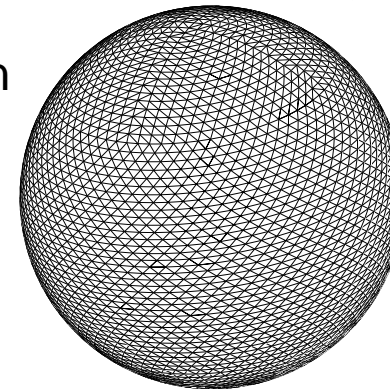


Atlas framework (Deconinck et al. 2017)

Hierarchical grids

- Adept to very high resolution
- Minimise data movement
- Support multi-grid approaches (e.g. Hotta and Ujje, 2018)
- Support data-driven parallel workflows (parallel data regions; NNs on the healpix sphere; chunking and lazy loading; ...)
- Healpix supports nested and ring ordering (still has latitudes!)
- Facilitate physical modelling including spectral and finite-volume

HEALPix grid



From Gorski et al. 2005

DestinE output data choice

Trends in WGNE member contributions

1. GPU adaptation / IO optimisation / single precision / new grid or grid partitioning on the sphere reported by several groups
2. Earth system approach: working on coupling techniques , ocean GPU porting, chemistry schemes difficult to port to GPU ; ...
3. Shift to AI and hybrid modelling (do we need to accelerate the physical model with AI on-demand forecasts ?)
4. Overall cost & time to solution remain relevant metrics for physical models
5. Significant time spent on HPC upgrades ; tales of adapting to new platforms
6. Geographically distributed compute & data flows

Annex Member Slides



CMC HPC/Exascale Projects: Background

- The move to consider exascale models has slowed because of increasing investments (both human and computational) in AI
- Given expected computing growth, the CMC is very unlikely to hit “exascale” in the next 15 years, making any associated planning very uncertain
- One thing that does seem clear is that if we hit exascale – even with physically based models – it will be using GPUs rather than CPUs
 - Additional resources need to be devoted to GPU-ready implementations of physical models well before exascale is achievable at the CMC

CMC HPC/Exascale Projects: Science

- Planned upgrades to the existing modelling system:
 - SLIMEX : semi-Lagrangian Implicit-Explicit time integrator
 - Combine SL and IMEX BDF2 time integrator
 - Second order in time, no extra off-centering and only one elliptic solve per step
 - Single-core performance evaluation and optimization for the physics
- Development of new algorithms:
 - Moving from a Yin-Yang grid to a rotated cubed-sphere grid
 - A space-time tensor formalism is used to express the equations of motion covariantly
 - The spatial discretization with the direct flux reconstruction method
 - New multistep exponential and implicit/Rosenbrock time integrators
 - Low-synchronization matrix-free Krylov solver
- Building a network of academic collaborators for the development of a hybrid physical/AI NWP system
 - Establish a close working relationship with the universities and private sector – joint projects in the next 5 years.
 - Goal: explore the spectrum of possibilities in applying numerical methods and ML to develop an optimal hybrid NWP model that will use the best of the two worlds.
 - Develop approaches in high-performance computing – the best numerical algorithms on today's supercomputer could be suboptimal in the future – and work with GPUs.

CMC HPC/Exascale Projects: Infrastructure

- Development of a new non-blocking IO server to solve increased IO bottleneck
- New more efficient MPMD multi-model coupling system
- Update to internal data format to enable parallel IO and multiple compression scheme allowing higher data compression
- Enable efficient check pointing on all model suites (standalone/coupled)
- Consider the GPU-heavy needs of future AI-based “on demand” forecasts, which would remove some of the performance requirements from R&D applications

HPC/efficiency efforts at Météo-France



Towards a general use of single-precision (32 bits) in operational NWP systems.

1. Operational use in all AROME¹ operational systems (forecast component only)
2. Next steps: soon operational use
 - i. in all ARPEGE forecasts
 - ii. in all trajectories within the assimilation cycle
 - iii. later, in parts of assimilation whenever possible

Adaptation to hybrid processors with accelerators:

- Full time step of ARPEGE model ported on GPU (except the surface model SURFEX). Being now optimized
- Plans:
 - use Field_API interface for spectral transforms
 - ecRad ported with Loki (instead of ecRad ported manually)
 - progress on scripts (use Loki) & environment for compiling,
 - refactor IO and SPP
 - finalize semi-implicit
 - full time step of AROME model to be ported on GPU
- Work done in collaboration with ECMWF and ACCORD² partners (DestinE On Demand project phase 2), and within TRACCS³ (Transformative Advances in Climate modelling for Climate Services) French national programme (WP on new computing paradigms).

Preliminary work on FVM dynamical core (LAM version and PMAP softwa

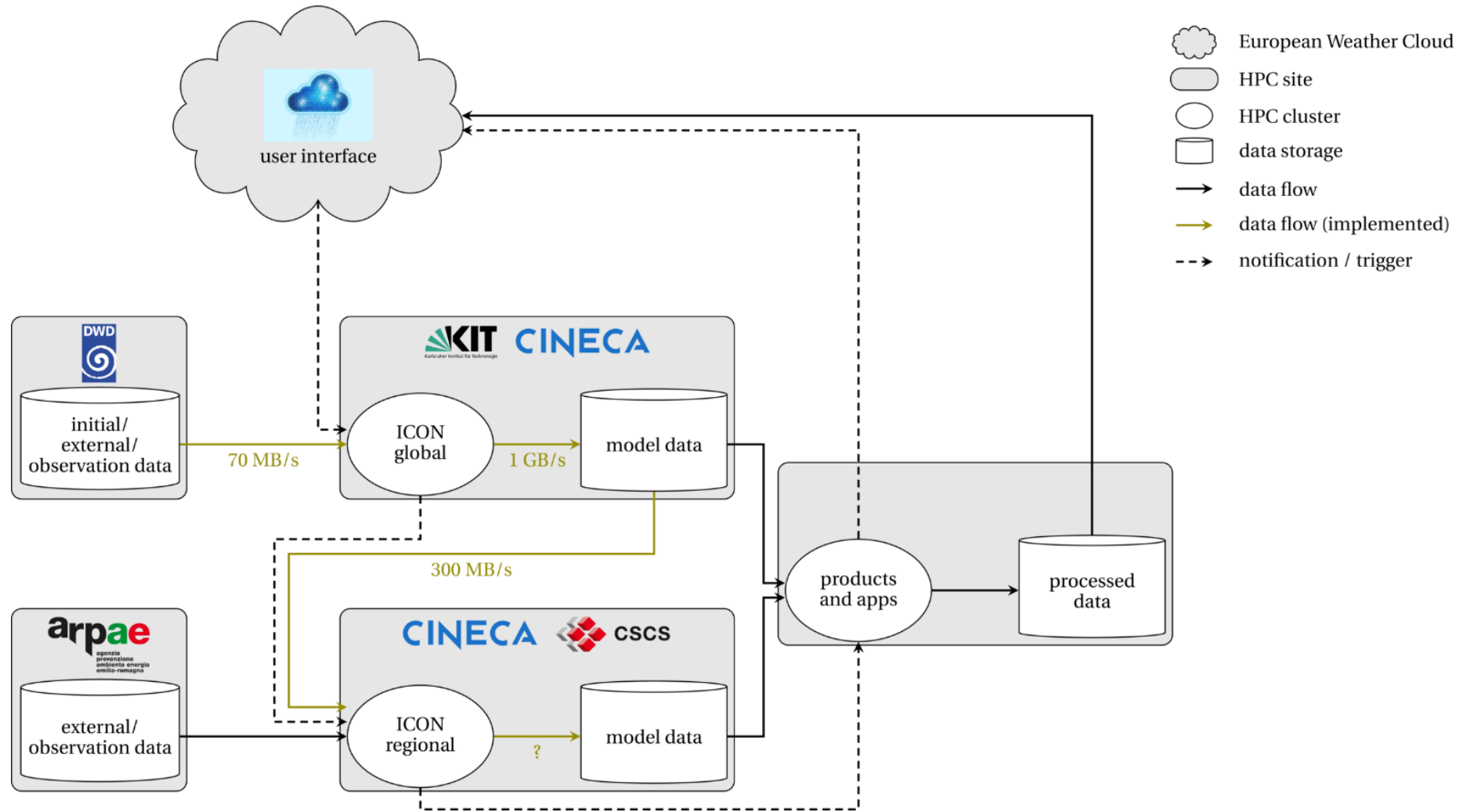
- The ICE3 microphysics scheme ported in Python for GT4Py DSL (DestinE On Demand project)

¹ Météo-France LAM NWP operational system

² [A Consortium for convective-scale modelling Research and Development](#)

³ [Transformative Advances of Climate Modelling for Climate Services](#)

Data flow within GLORI-DT



GLORI-DT on HoreKa (KIT)

HoreKa is a national supercomputer managed by SCC at KIT

- Blue: 570 nodes with 76 Intel Xeon cores/node (+ 40 high memory nodes)
⇒ 2.3 PFlop/s
- Green: 167 nodes with additional 4 NVIDIA A100-40 GPUs/node
⇒ 8.0 PFlop/s

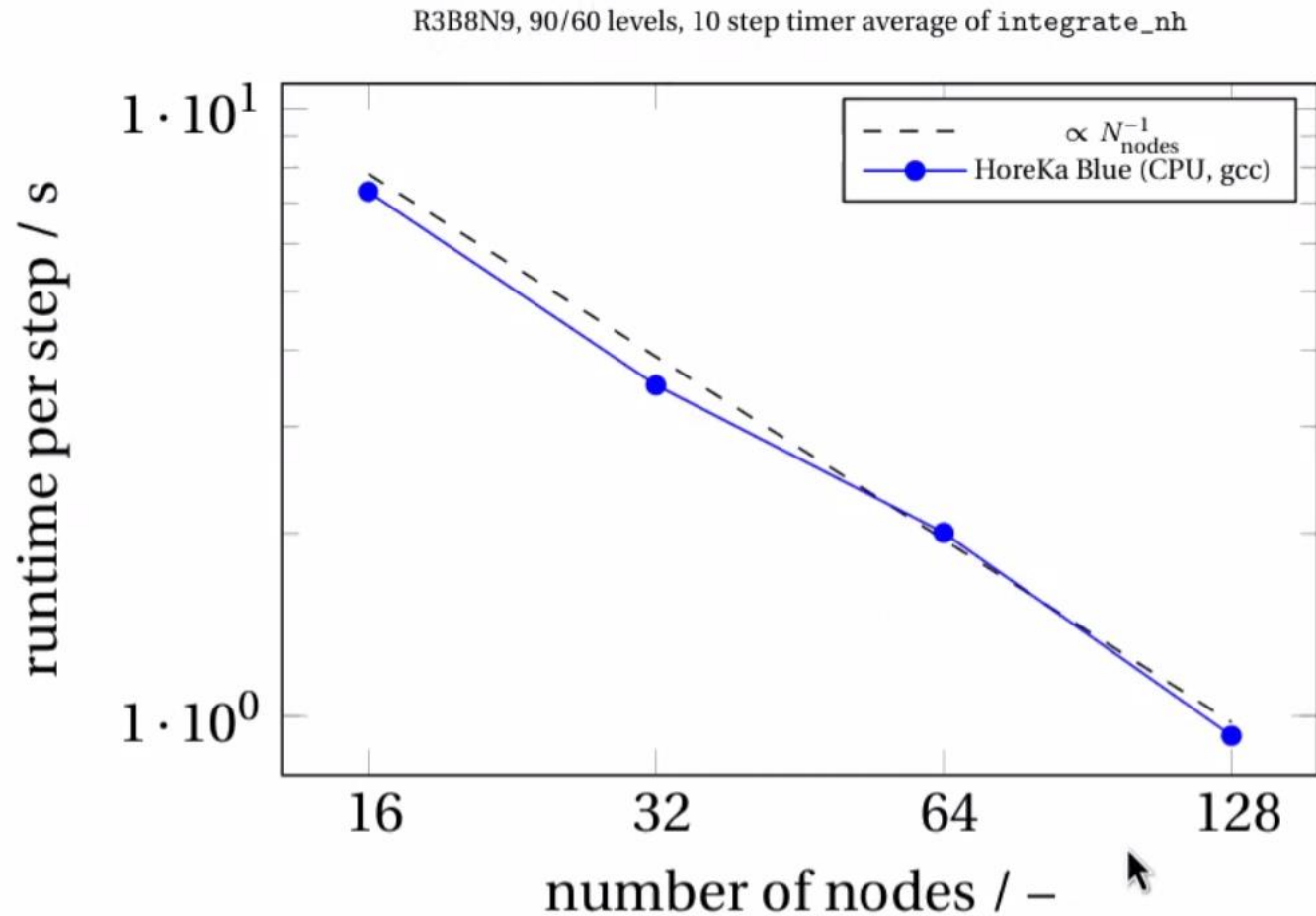
ICON support has been established with GLORI project

- Implementation of state-of-the-art config wrappers (including GPU support)
- Buildbot site since Nov 2023 (CPU & GPU builders)
- hk-project-glori-init: 12.3 MCPUh + 115 kGPUh resources – soon extended

Simple Storage Service (S3) available for data exchange

GLORI-DT on HoreKa (KIT)

- Technical test:
ICON global at 6.5 km with 3.25 km nest on Horeka on CPU
-> It scales well!



GLORI-DT on Leonardo (CINECA)

 **LEONARDO** is a European supercomputer managed by CINECA

- Booster: 3456 nodes with 32 Intel Xeon cores/node and 4 NVIDIA A100-64 GPUs/node
 ⇒ 238 PFlop/s

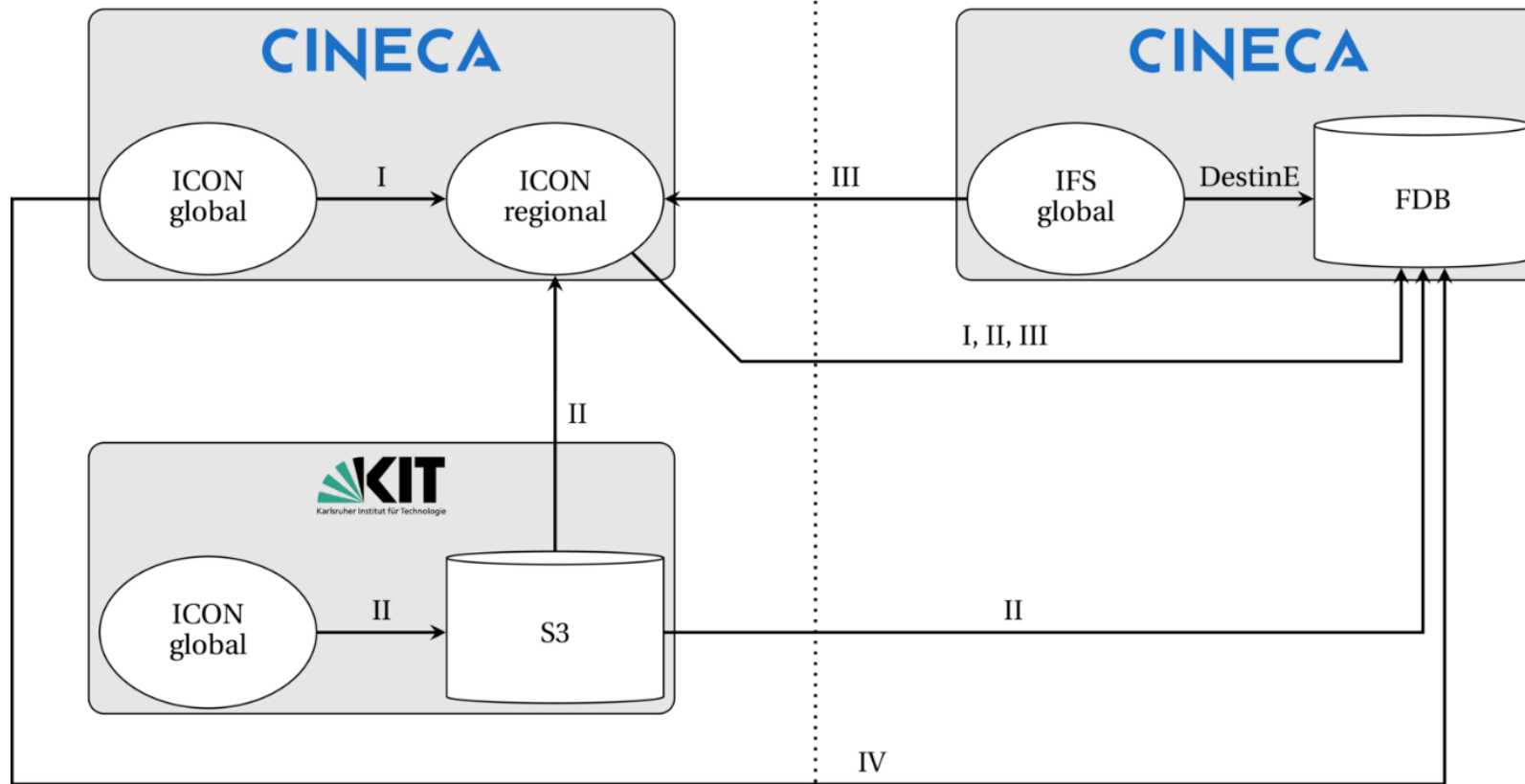
ICON support is currently being established

- ICON dependencies managed via Spack
- Build wrappers based on HoreKa setup have been implemented
- Passes basic tests, but rigorous testing still to be done

Leonardo plays also a key role in Destination Earth

- Interoperability via Fields DataBase (FDB)
- Developing the capability for the GLORI-DT to output to FDB via output coupler / YAC

Uniting the Twins: GLORI4DE



- HPC site
 - HPC cluster
 - data storage
 - data flow
- I,II,III,IV GLORI4DE scenarios

Other activities

- ICON-D05: additional deterministic 500-m forecasts for Germany (8x/day), accomplished by double two-way nesting into ICON-D2 (and starting from the D2 analysis)
- 48-hour forecasts are supposed to be completed within 30 minutes, which requires moderately strong scaling
- Easy to achieve on our NEC-SX Aurora (16 nodes VE30; approx. 5% of our HPC when ongoing upgrade is completed), but challenging on GPUs
- Thus, we are working together with our Swiss colleagues on improving the efficiency on GPUs
- Pre-operational since mid-September

- For future global convection-permitting configurations, ways to parallelize reading GRIB2 input data are explored (output has sufficient parallelization)
- Besides this, forecast quality issues at (global) convection-permitting resolution are investigated

NOAA Unified Forecast System

- **UFS components:** Atmos (fv3 dycore), Land (Noah-MP), Ocean (MOM6), Ice (CICE6), Wave (WAVEWATCH III), Aerosol (GOCART), Air Quality (CMAQ), CMEPS mediator, CCMPP physics
- **UFS Applications:**
 - **Global:** GFS (medium-range NWP), GEFS (ensemble), SFS (seasonal), UFS-aerosol, [Whole Atmosphere Model \(WAM\) for Space Weather Prediction](#)
 - **Regional:** HAFS (hurricane), RRFS (regional NWP), Online-CMAQ (air quality), [Atmospheric River \(AR\)](#).

Improvement for I/O and computational efficiency

- Parallel NetCDF with data compression applied to history files, [and expanded to hurricane moving nests](#)
- ESMF managed threading -- apply different threads for different UFS components
- Single and double precision dycore
- 32-bit physics (selected parameterization suites)
- [Exchange grid capability](#)
- [Testing Zstandard data compression](#)
- [Move I/O to the NUOPC layer as a component of the earth system model, to be shared by all UFS components.](#)

HPC upgrade

- **OLD:** WCOSS2, CRAY EX, 2560x2 nodes, 327Kx2 cores, 12,100 x2 TF peak performance.
- **New as of Aug 2023:** WCOSS2, CRAY EX, 3060x2 nodes, 392Kx2 cores, 14,400 x2 TF peak performance.

On the Cloud

- [Running experimental hurricane ensemble forecast \(HAFS\) and regional high-res ensemble forecast \(RRFS\) on the Cloud.](#)

WCOS2 In Operation Since August 2023

Locations

- Manassas, VA
- Phoenix, AZ

Performance Requirements

- 99.9% Operational Use Time
- 99.0% On-time Product Generation
- 99.0% Development Use Time
- 99.0% System Availability

Configuration

- Cray EX system
- **14.4 PetaFlops**
- Multi-tiered storage
 - 2 flash filesystems each with...
 - 614 TB usable storage
 - 300 GB/s bandwidth
 - 2 HDD filesystems each with...
 - 12.5 PB usable storage
 - 200 GB/s bandwidth
 - Total aggregate - 26.2PB at 1TB/s
- Lustre parallel filesystem
- PBSpro workload manager
- Eclflow scheduler

- Compute nodes
 - **3,060 nodes (60 spare)**
 - 3391,680 cores
 - **128 cores/node**
 - 1.3 PB of memory
 - 512 GB/node
- Pre/post-processing nodes
 - 132 nodes (4 spare)
 - 8,448 cores
 - 64 cores/node
 - 132 TB of memory
 - 1TB/node
- 200Gb/s Slingshot interconnect

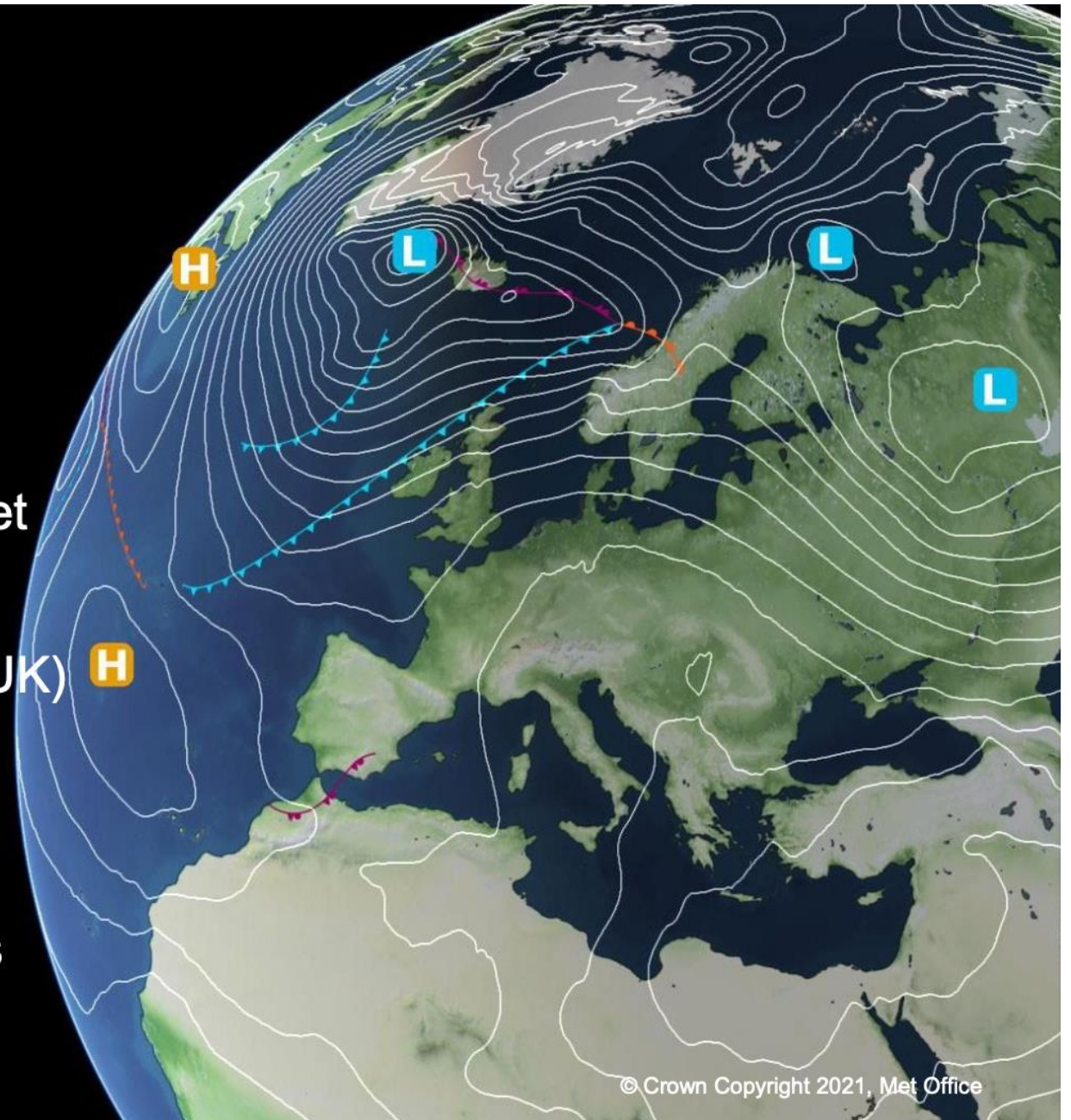
Recent work on NEMO optimisation

D. Calvert, A. Aguiar, M. Bell,
I. Kavcic, C. Maynard, H. Shepherd (Met
Office, UK)

C. Dearden, A. Porter, S. Siso (STFC, UK) 

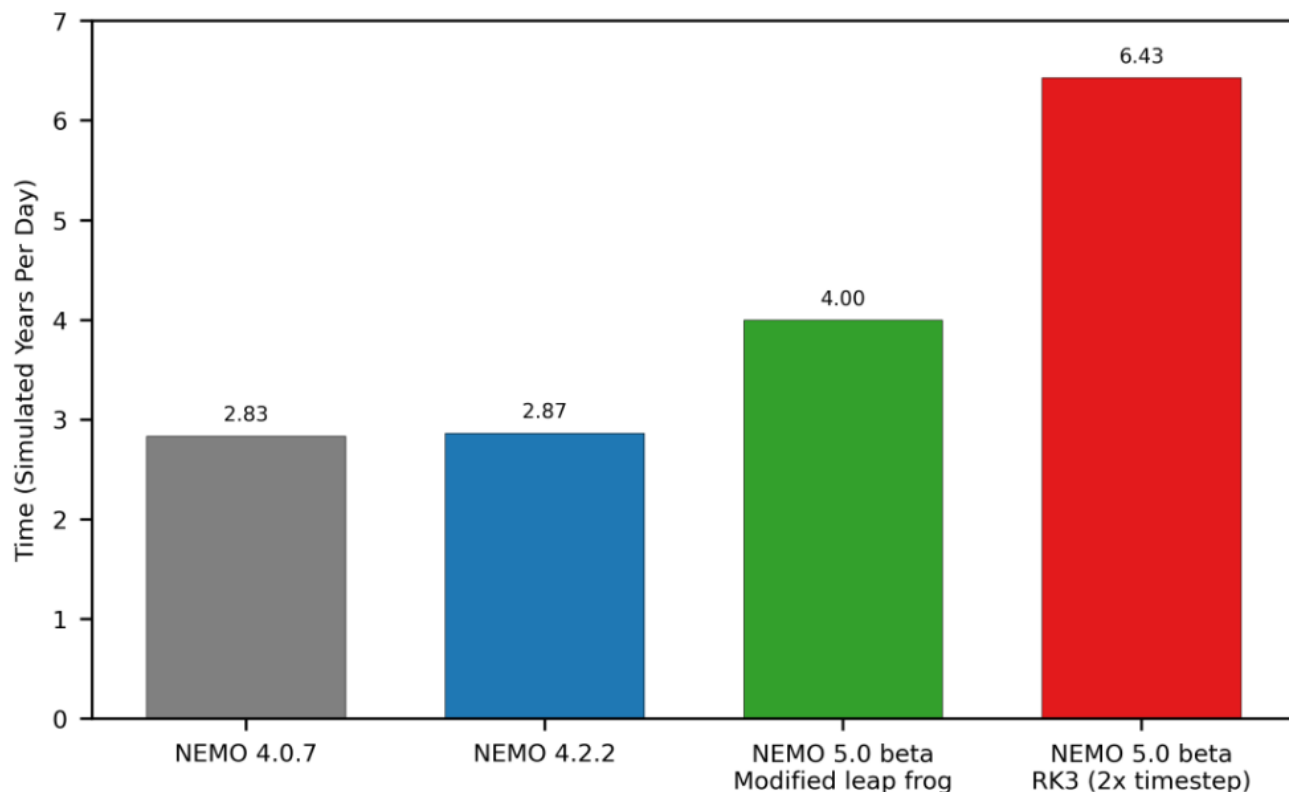
T. Deakin, K. Olgu, A. Sadawarte, G.
Williams (University of Bristol, UK)

Various NEMO systems team members



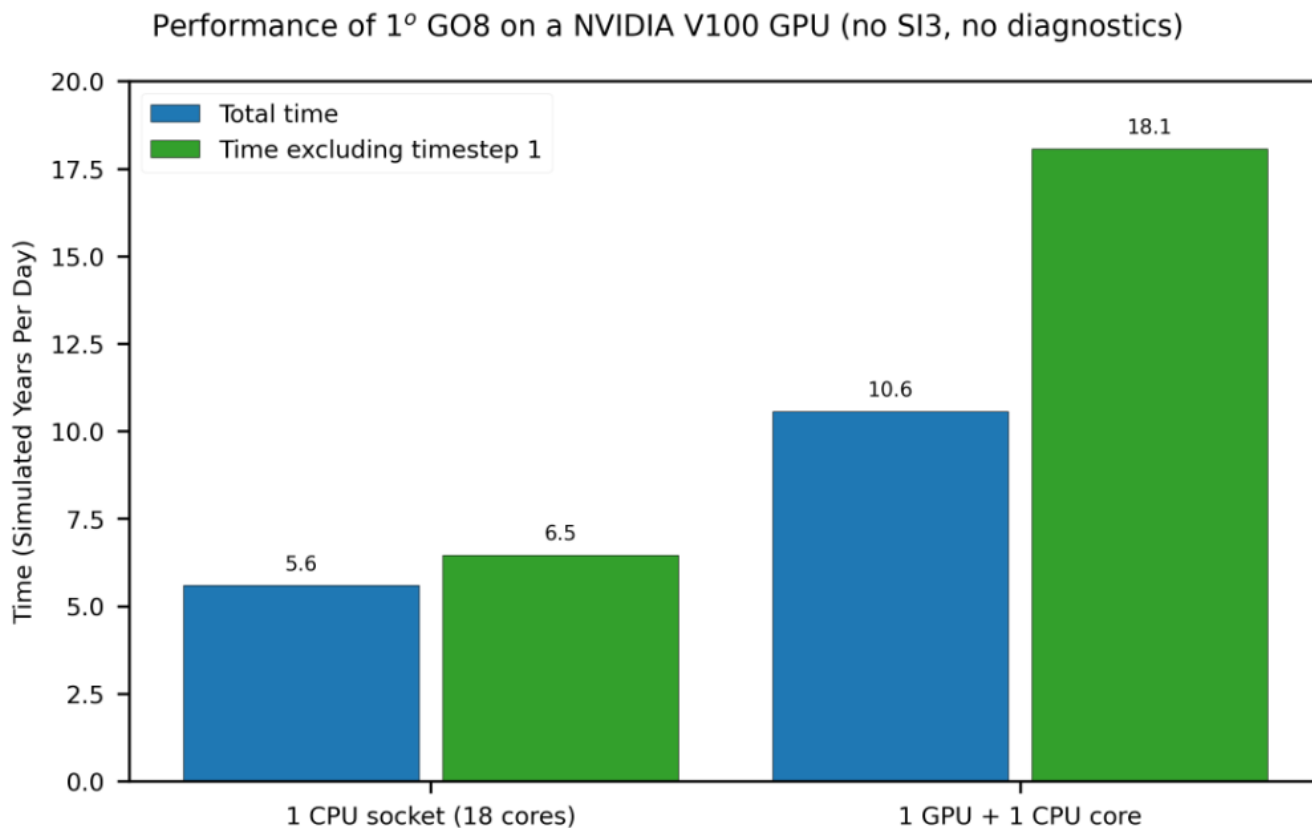
NEMO performance on CPUs

Performance of 1/4° "GOSI10-like" simulations



- Run time of 2-month 1/4° NEMO-SI3 (ocean & sea ice) simulations in SYPD
- NEMO 5.0 introduces **several optimisations for CPU architectures**
 - Reduced number of communications
 - Added delayed communications
 - Reduced memory footprint of code
 - Loop tiling
- The new **3rd order Runge-Kutta (RK3)** time-stepping scheme is more stable and allows the time step to be increased

NEMO performance on GPUs



- [PSyclone](#) is used by NEMO and several other models to transform their codes for running on GPUs
- STFC's work for the [ExCALIBUR project](#) demonstrated that an ocean-only simulation based on NEMO 4.0.2:
 - Runs ~3.4x faster on a NVIDIA V100 GPU than on a 16-core Intel Skylake CPU (1° grid)
 - Runs on 90 NVIDIA V100 GPUs with performance equivalent to 270 16-core Intel Skylake CPUs (1/12° grid)
- Recent work under the [Met Office NGMS programme](#) achieved a similar (~2.8x) performance increase for NEMO 4.0.4
 - 1 NVIDIA V100 GPU vs 1 18-core Intel Cascade Lake CPU
- This is currently being repeated for NEMO 5.0



HPC readiness: Input from JMA

Japan Meteorological Agency

Outline

- Three main JMA models are being prepared for future HPCs
 - GSM (JMA's operational global model)
 - ASUCA (JMA's operational regional model)
 - MRI.COM (JMA/MRI's operational/research ocean model)
- This ppt reports current status and plans of the readiness.
 - Reduced precision version models
 - GPU porting

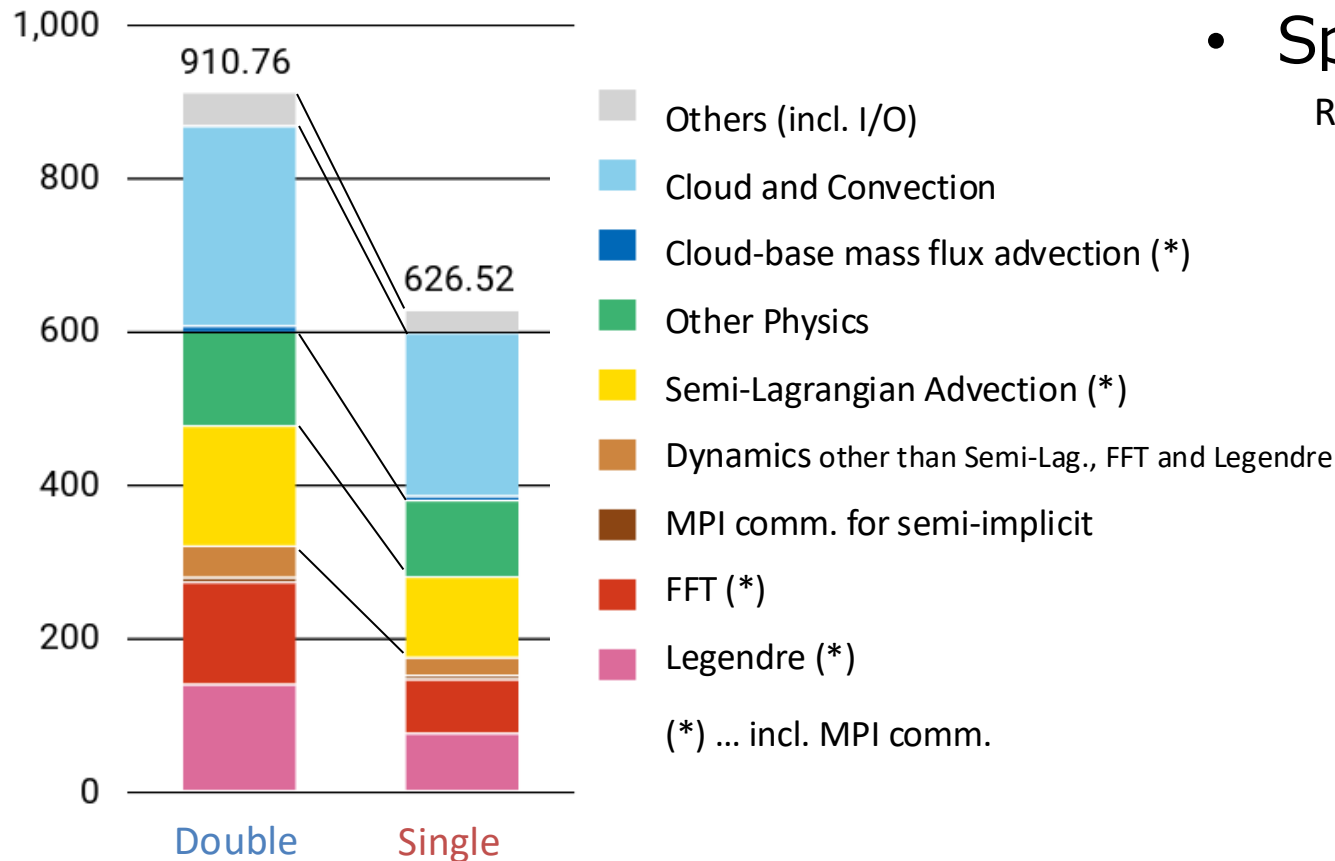
Development of reduced precision models

- GSM and ASUCA
 - Implemented switch functions between single/double precisions
 - Speed-up by 30% in the both models with the single precision mode, further speedup (~40%) is aimed.
 - Fixed model failure and significant numerical accuracy degradation issues, mainly due to loss of digits, information drop and overflow/underflow etc...
 - TIPS and know-how accumulated
- MRI.COM
 - Mixed precision approach is tested
 - Several variables (e.g., mass and volume) need to be kept as double precision to obtain both speed up and accuracy

```
integer(4) :: WPR = kind(1.0d0) ! double precision
!integer(4) :: WPR = kind(1.0e0) ! single precision
real(kind=WPR) :: some_data

some_data = 1.0_WPR
```

Single precision GSM



MPI rank average elapsed time [s] of Tq959 GSM (dx~13km) for 132hr time integration

49nodes, 390ranks (incl. 6 I/O ranks) and 14 OpenMP threads on Fujitsu PRIMAGY CX2550 M7, Intel Fortran(2021.9.0) with "-O2" option

- Speed-up by **30%**

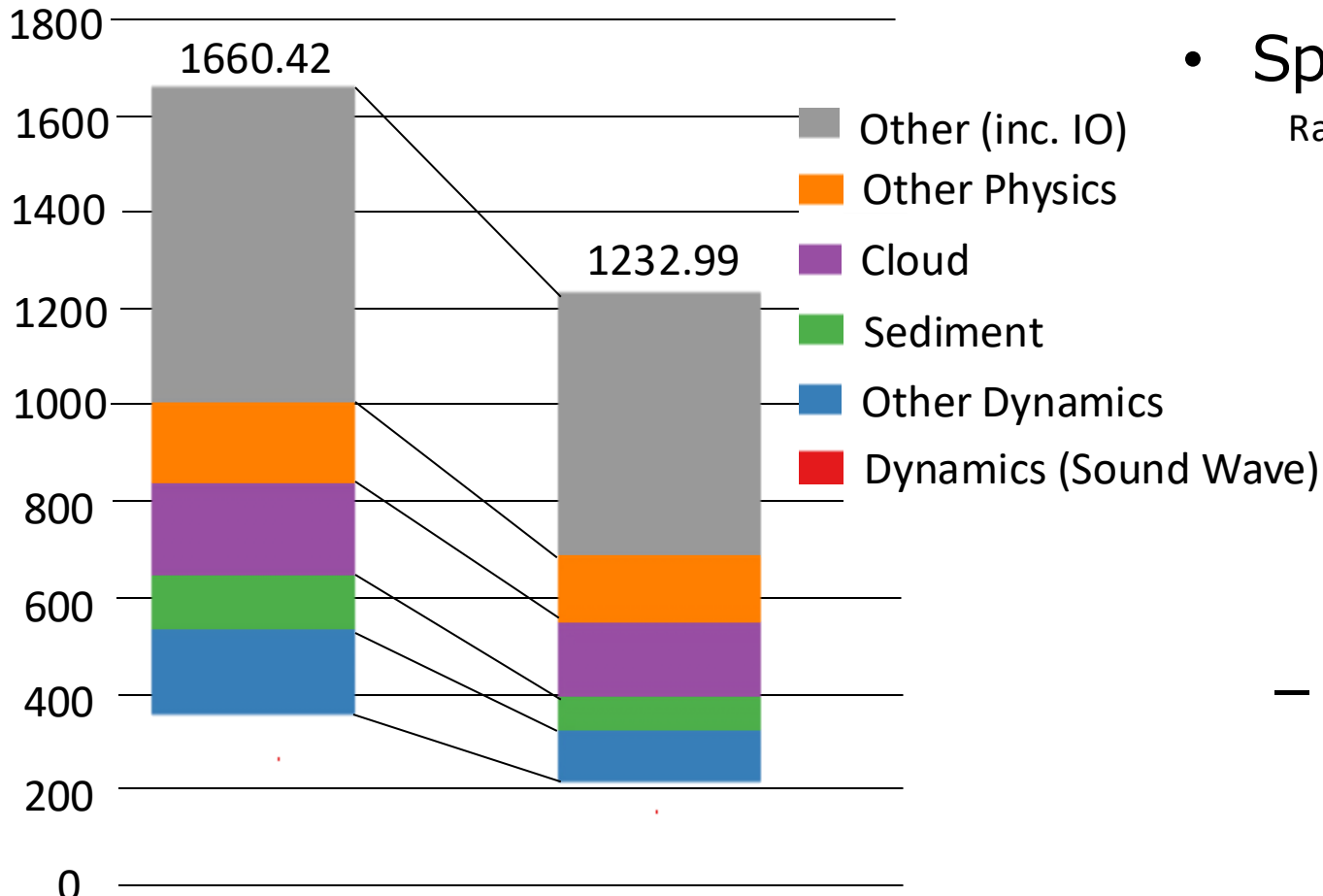
Ratio of elapsed time in single precision GSM to double precision

Process	Single/Double
Others (incl. I/O)	0.67
Cloud and Convection	0.82
Other physics	0.81
Semi-Lagrangian Advection	0.67
Other dynamics	0.54

Small speed-up rates in physics parameterization, particularly in specific subroutines presumably due to:

- Slow convergence in iterative algorithms
- SIMD suppression in loops with complex "if" branches

Single precision ASUCA



- Speed-up by **30%**

Ratio of elapsed time in single precision asuca to double precision

Process	Single/Double
Others (incl. IO)	0.83
Other Physics	0.84
Cloud	0.81
Sediment	0.62
Other Dynamics	0.61
Dynamics (Sound Wave)	0.60

MPI rank average elapsed time [s] of asuca (configuration of LFM ,dx 2km) for 10.5hr time integration

19nodes, 304ranks (incl. 16 I/O ranks) and 14 OpenMP threads
on Fujitsu PRIMAGY CX2550 M7, Intel Fortran(2021.9.0) with "-O2" option

- Small speed-up rates in physics parameterization, particularly in specific subroutines presumably due to:

- SIMD suppression in loops with complex "if" branches

Issues and remedies in transition from double to single precision

- **Issue:** zero-division at time-step mean solar zenith calculation (based on Hogan and Hirahara, 2016) : The issue occurs at sunrise at the end / sunset at the start of a timestep
- **Remedy:** use an approximated form without near zero-division
- Note that this issue is atmospheric state-independent (only dependent on date, spatial and time resolution), thus predictable whether / when / where a model fails

Before

$$\overline{\cos \theta} = \sin \delta \sin \phi + \frac{\cos \delta \cos \phi (\sin h_{\max} - \sin h_{\min})}{h_{\max} - h_{\min}}$$

In a model failure case:

$$h_{\max} - h_{\min} = 0 \quad \text{in single precision (zero due to loss of digit)}$$

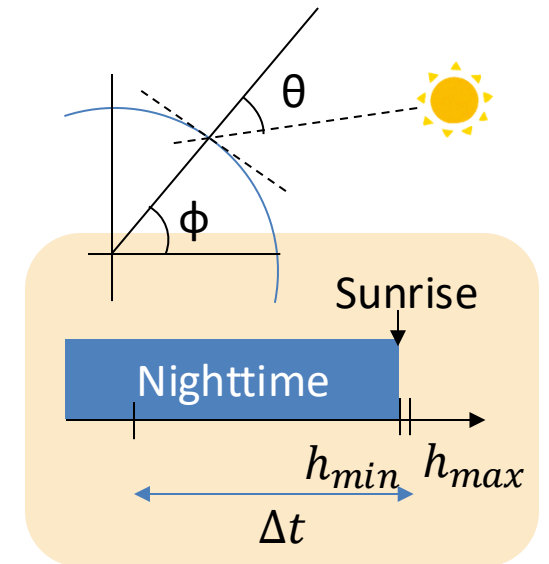
note: minimum value* in double precision: $h_{\max} - h_{\min} \sim O(10^{-12})$

*... Tq959 (~13km), 24-hour forecast, cases for the next few years

After

$$\overline{\cos \theta} \sim \sin \delta \sin \phi + \cos \delta \left[\cos \phi \cos (h_{\max} - h_{\min}) \right]$$

$$\text{if } h_{\max} - h_{\min} < \epsilon$$

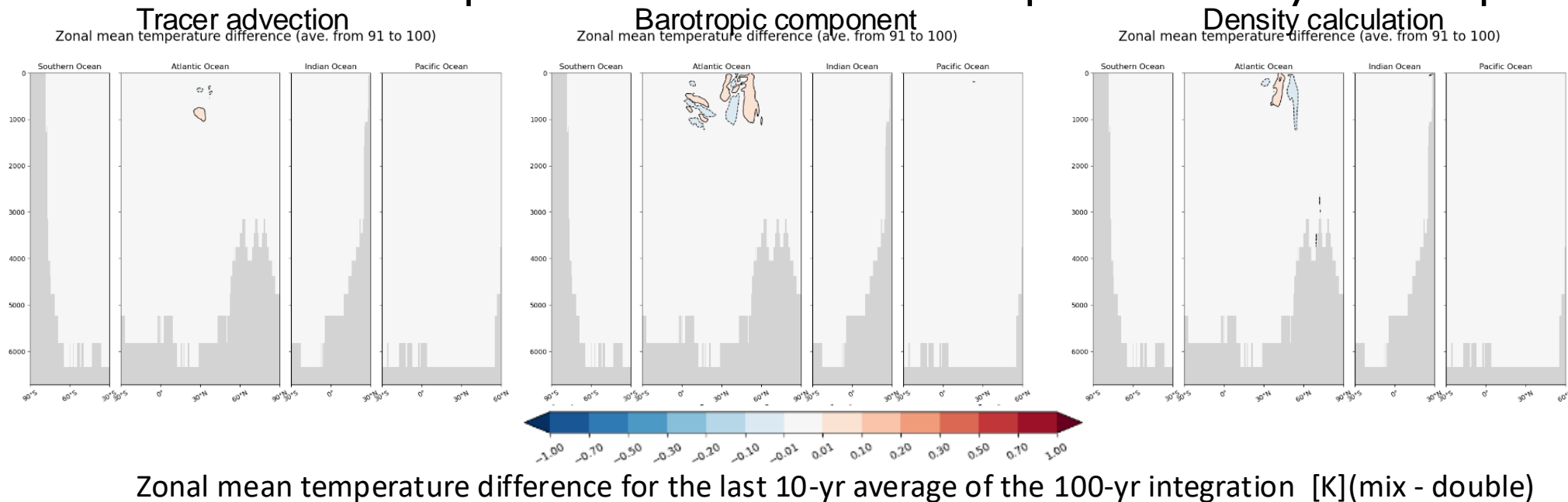


δ : Declination ϕ : latitude
 h_{\min} and h_{\max} : start and end of
 daytime in a time step

(ϵ is set by considering numerical safety and accuracy of the approximation, 10^{-7} for double precision, to be set for single precision)

Mixed precision in MRI.COM

- Mixed precision approach: Several variables (e.g., mass and volume) need to be kept as double precision to obtain both speed up and accuracy
 - Tracer advection, momentum equation for barotropic component, density calculation
- To be finished implementation of mixed precision by next spring

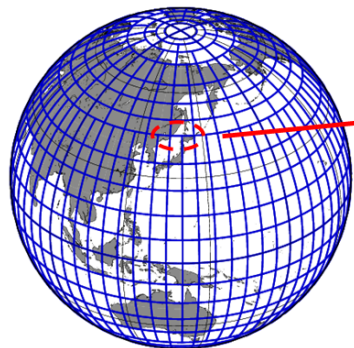


Zonal mean temperature difference for the last 10-yr average of the 100-yr integration [K](mix - double)

GPU porting: Status and plans

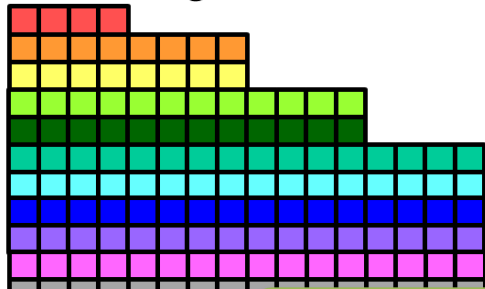
- GSM
 - most of processes (parallelization, spectral transform, dynamics, cloud and convection) have been ported to GPU. To be completed porting by next spring
 - For GSM, full-GPU appears to be better rather than offloading only for bottlenecks, as optimum data structure for OpenMP (CPU) and OpenACC (GPU) parallelization is different.
- ASUCA
 - GPU porting completed (except I/O). Preliminary results were reported to WGNE-38.
 - Further evaluation and optimization are ongoing.
- MRI.COM
 - Several bottlenecks (tracer-advection, momentum equations for barotropic and baroclinic components) have been ported.
 - To be completed porting in a few years

Tq959L128 960mpi



Lon. →
Lat. ↓

Horizontal grid boxes in a MPI process



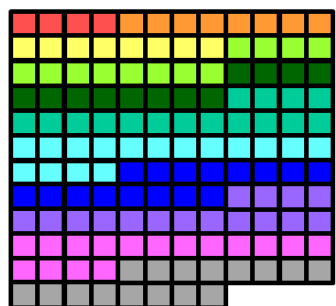
NUMI_I=12

NUMI_I=36

Max array size of array in the "i" direction is controlled by a parameter "NUMI_I"

For CPU with OpenMP
(larger outermost loops
for thread parallelization)

For Vector machines or GPU with OpenACC:
(larger innermost loops for vectorization)



i →
j ↓

NUMI_I

NUMJ_S

Typical source code in GSM

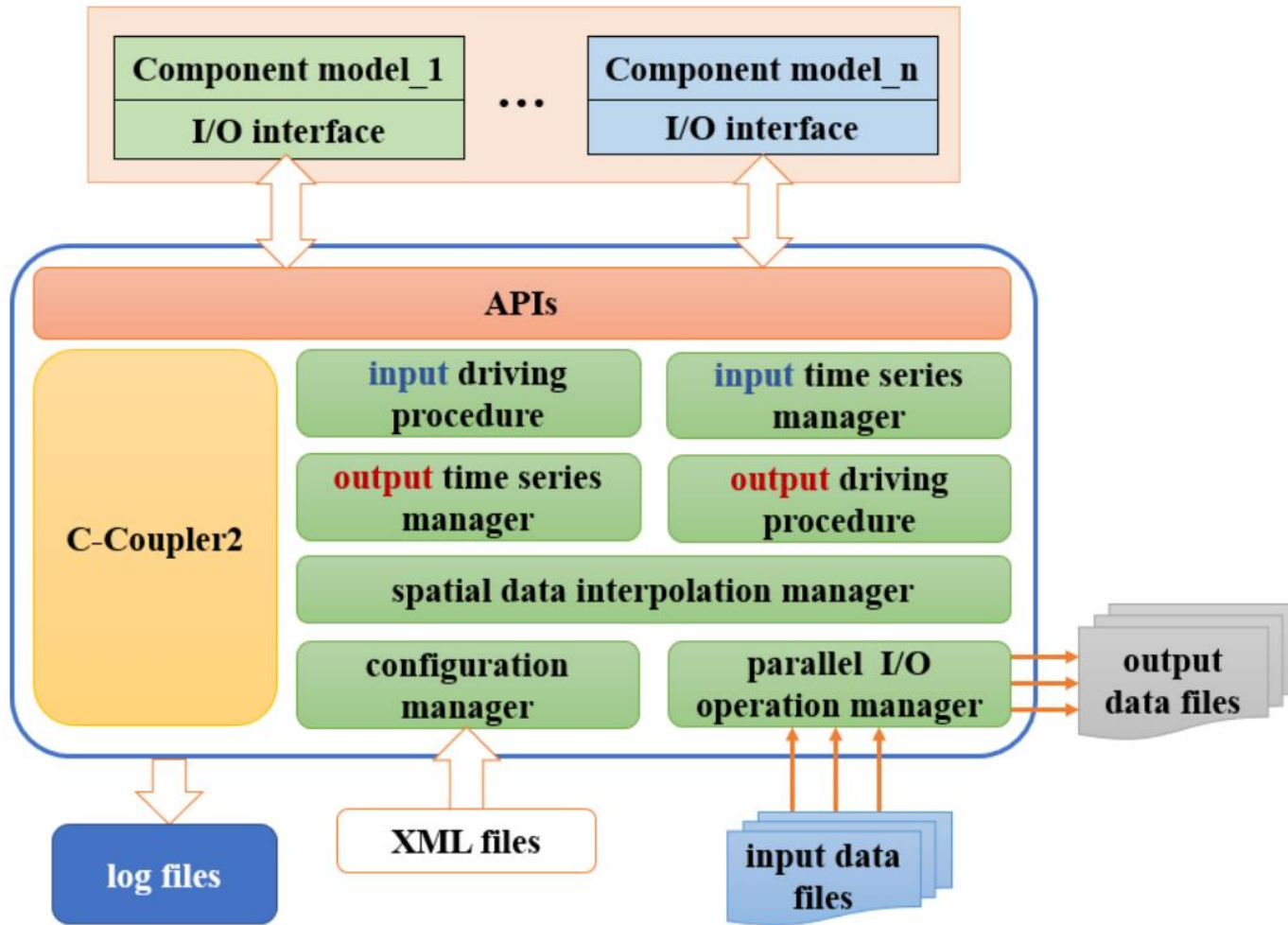
```
integer(4) :: WPR = kind(1.0d0) ! double
real(kind=WPR), dimension(NUMI_I, NUMFA_I, NUMJ_S) ::
array1(:,:,:),array2(:,:,:)
!$OMP PARALLEL default(SHARED), private(i,k,j)
!$OMP DO schedule(DYNAMIC)
do j = 1, NUMJ_S
!$acc kernels &
!$acc present(NUMI,array1, array2,...)
do k = 1, NUMFA_I
do i = 1, NUMI(j)
array1(i,k,j) = "parallel calculation using array2(l,k,j)"
end do
end do
!$acc end kernels
end do
!$OMP END DO
!$OMP END PARALLEL
```

Elapsed time [s] of Tl159L128 GSM (~110km) for 6hour time integration (1node, 8MPI, 14threads, 8GPU)

	All CPU (optimized outermost loop for CPU(OpenMP))	Semi-Lag. on GPU, others on CPU (optimized innermost loop for GPU(OpenACC))
Semi-Lagrangian advection	2.9413	1.5076 😊
Cloud and convection	1.0733	13.881 😞
Other physics	2.3951	20.433 😞

Optimized for GPU, but
awful performance in CPU

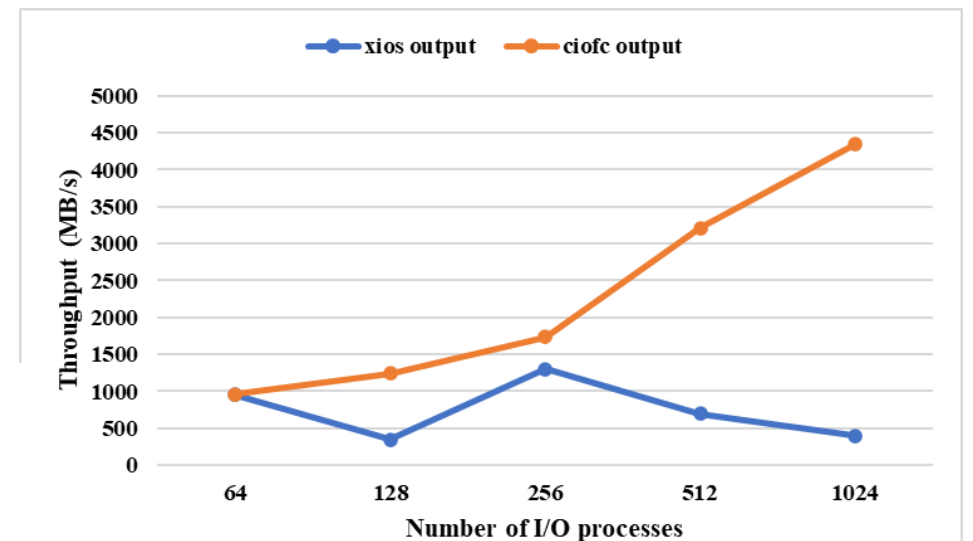
CIOFC: Common Input/Output Framework based on C-Coupler



Coupling in MCV model, CMA, China

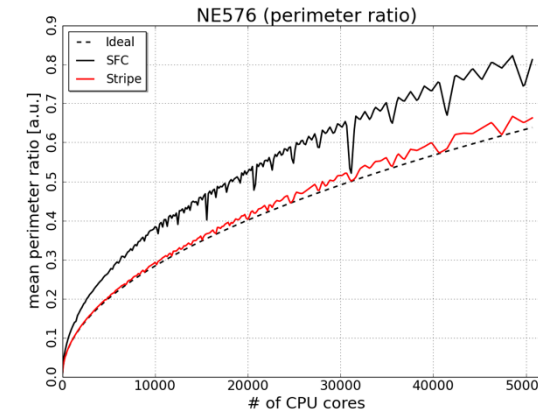
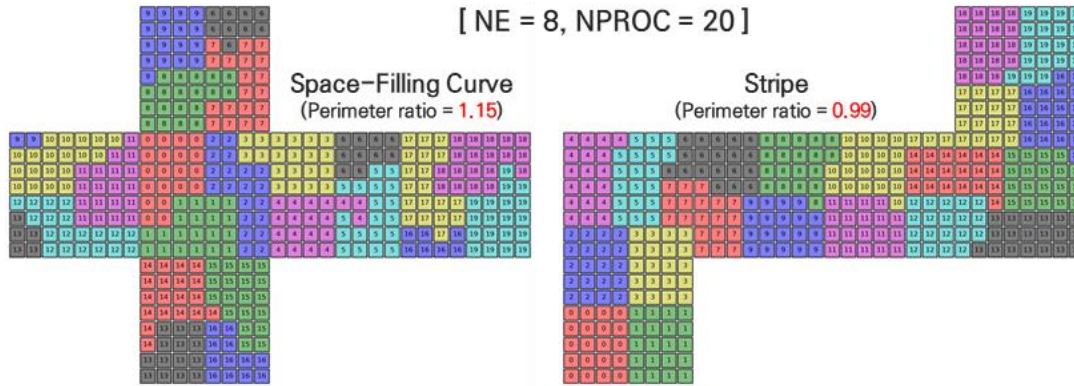
At a global 10km resolution, with 80 levels and floating point data type

- **Adaptively input time-series dataset with spatial and time interpolation**
- **Automatically output data aperiodically or periodically with spatial interpolation**
- **Efficient synchronous parallel input/output**



New grid partitioning “Stripe method”

: significantly reduces the model's communication amount (communication \propto perimeter rate)



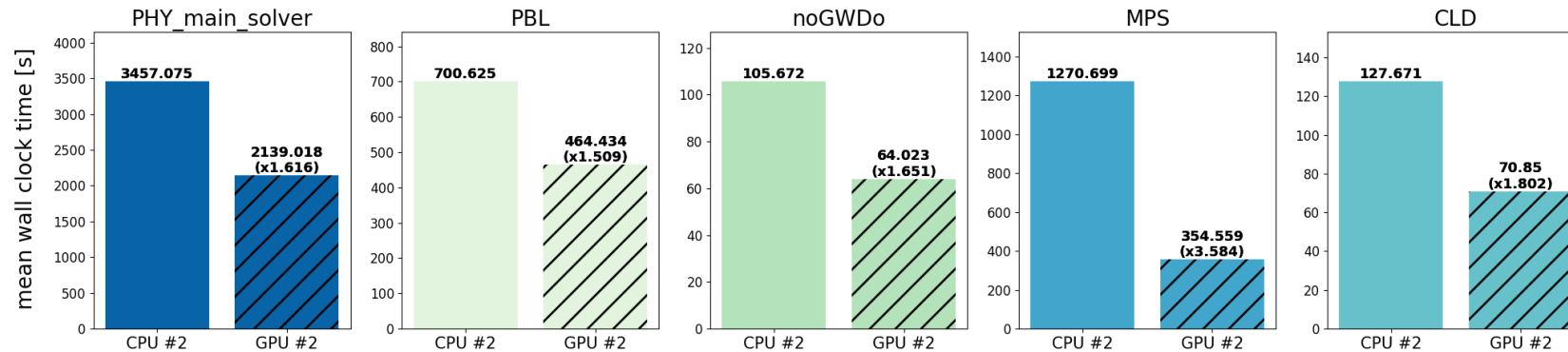
[Space-Filling Curve (old) vs Strip method (new) in grid partitioning]

- KIM: Korean Integrated Model developed by KIAPS using Cubed-sphere grid

GPU porting for KIM physics

: achieved about 1.6x speed-up when using the GPU

CPU: Intel Xeon Broadwell E5-2620v4 x 2, GPU: NVIDIA V100 x 2

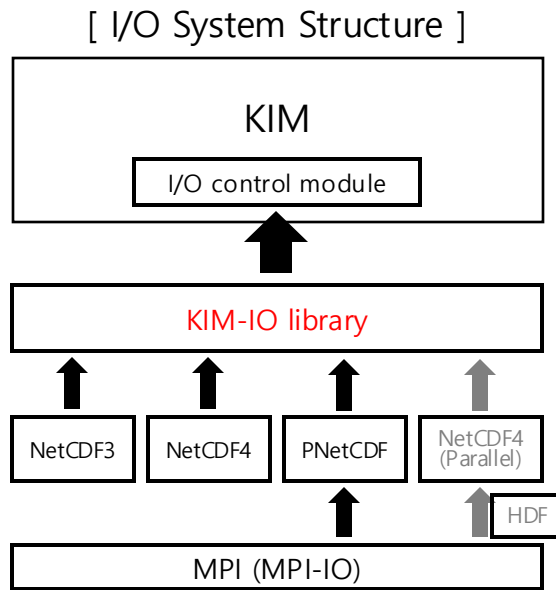


I/O optimization (KIM-IO)

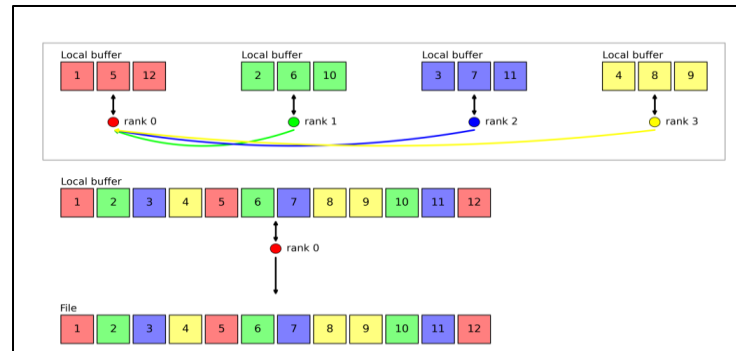
- KIM-IO: an I/O library for KIM developed by KIAPS

A new option for I/O decomposition by adopting the rearrange method used in PIO (Parallel-IO)

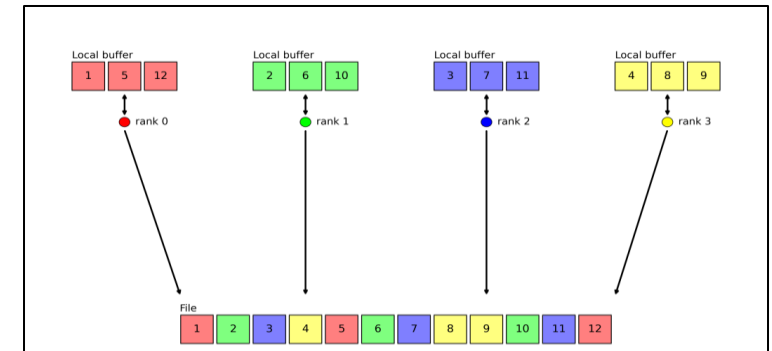
Currently, expanding the use of KIM-IO to support all components of KIM prediction systems (OPS, DA, Ocean model..)



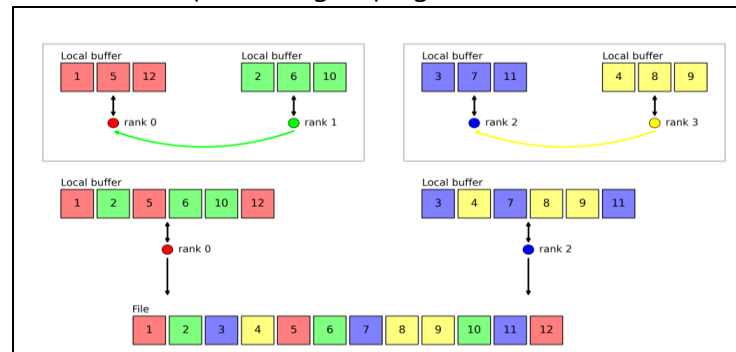
(a) Serial I/O



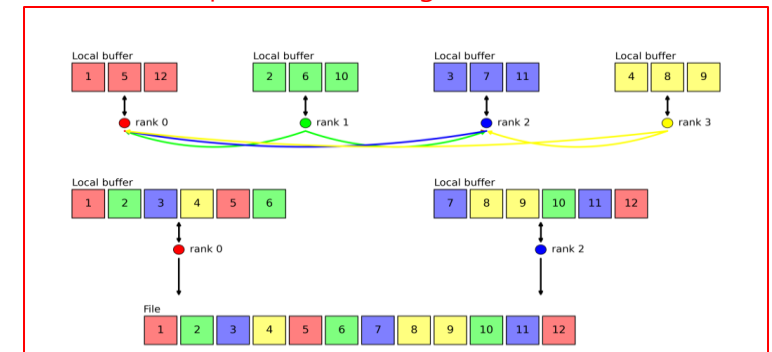
(b) Parallel I/O



(c) I/O decomposition: grouping



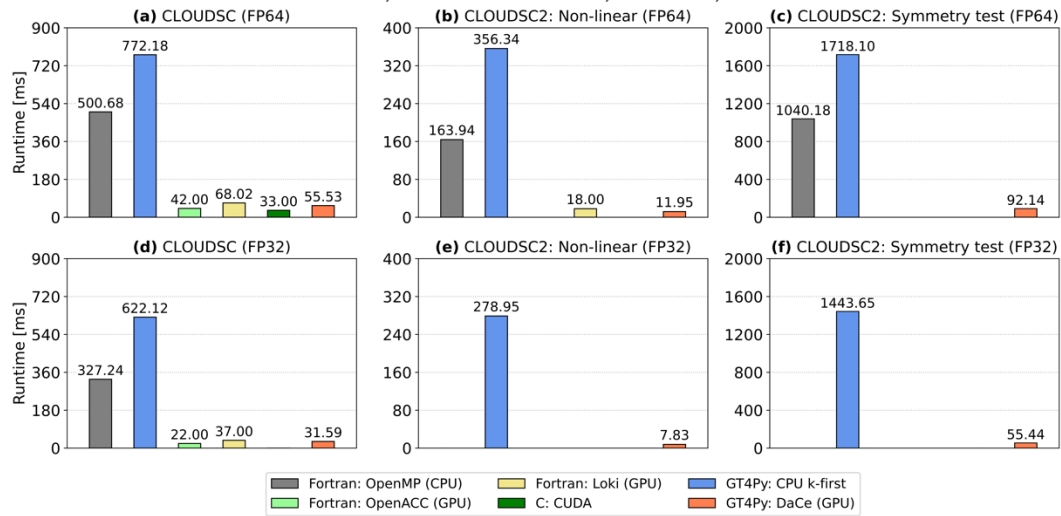
(d) I/O decomposition: rearrange



Various options in I/O methods used in KIM. Below two display the I/O decomposition methods in KIM-IO
 (d) is a new I/O decomposition using the rearrange method

Exploring GT4Py for the NWP domain using ECMWF microphysics schemes

Piz Daint, Nvidia P100, CSCS, Switzerland



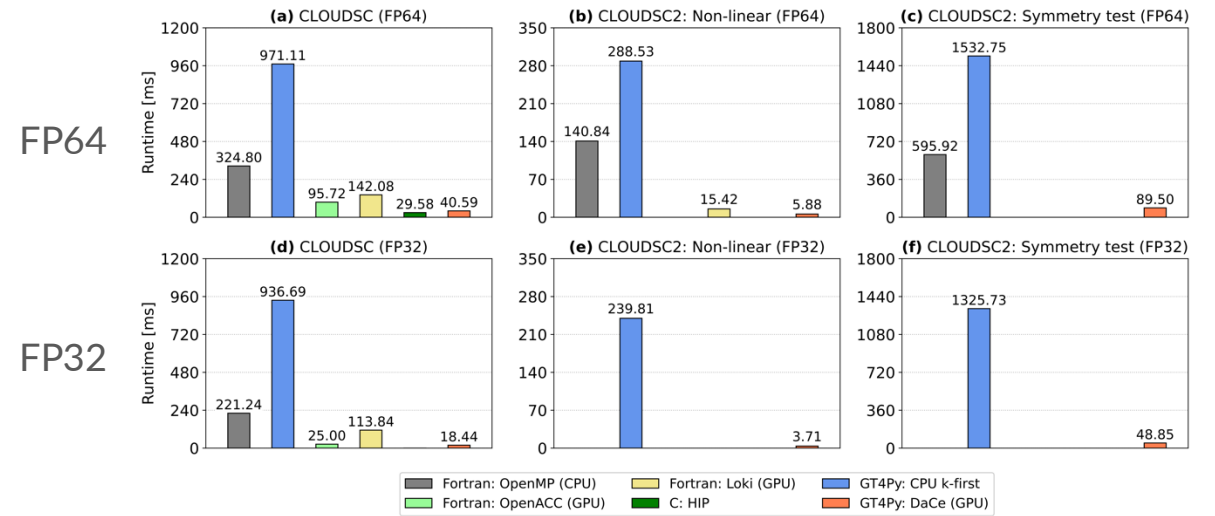
CLOUDSC

simplified nonlinear

TL/AD symmetry test

- Performance testing Python implementations of CLOUDSC, simplified nonlinear CLOUDSC2, tangent-linear CLOUDSC2, and adjoint CLOUDSC2 with GT4Py
- GPU vs CPUs, 64-bit vs 32-bit precision, various GT4Py backends
- Ubbiali et al. (<https://gmd.copernicus.org/preprints/gmd-2024-92/>)

LUMI-G, AMD MI 250X CSC, Finland



FP64

FP32

MeluXina, Nvidia A100, LuxConnect, Luxembourg

