Preparing IFS for HPC accelerators via source-to-source translation

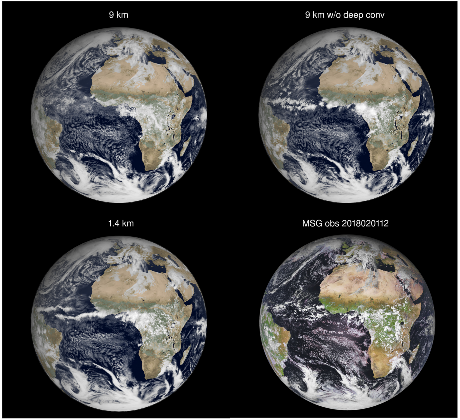
M. Lange, B. Reuter, O. Marsden

michael.lange@ecmwf.int

European Centre for Medium-Range Weather Forecasts

Towards km-scale global models

- Seasonal global simulation at 1.4 km horizontal resolution (INCITE) [1]
- 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



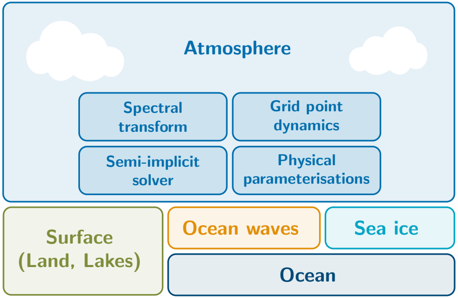
[1] Nils P. Wedi et al. "A Baseline for Global Weather and Climate Simulations at 1 km Resolution". In: *Journal of Advances in Modeling Earth Systems* 12.11 (2020), e2020MS002192. DOI: `https://doi.org/10.1029/2020MS002192`.

Hybrid 2024 - Preparing IFS for HPC accelerators

Hybrid 2024: Internal 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 to target different programming models / compiler toolchains
- Continuous testing on available hardware

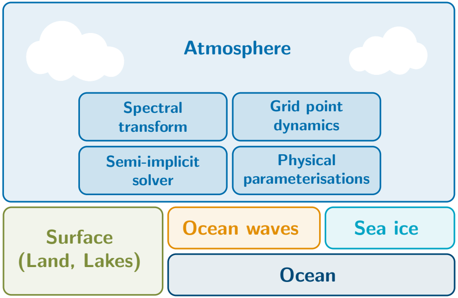


Hybrid 2024 - Preparing IFS for HPC accelerators

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

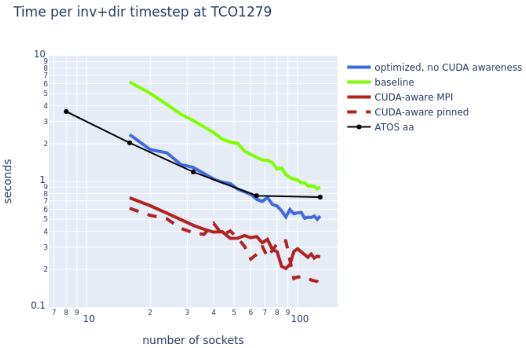


ECTRANS - An open-source Spectral Transform library

- Extracted Spectral Transform routines and packaged as stand-alone project (/ecmwf-ifs/ectrans)
- No dependency on ifs libraries; minimal requirements stand-alone FIAT library (/ecmwf-ifs/fiat)

GPU-enabled version available

- GPU-enabled (NVIDIA) branch is now in the repo
- Fortran + OpenACC + vendor-specific libraries (FFTs rely on CUFFT, GEMM calls on cuBLAS)
- Good scaling behaviour; uses CUDA-aware MPI (GPU-to-GPU MPI communications via NVLink)
- **WARNING: This is work in progress! Not all features are supported on GPU yet!**



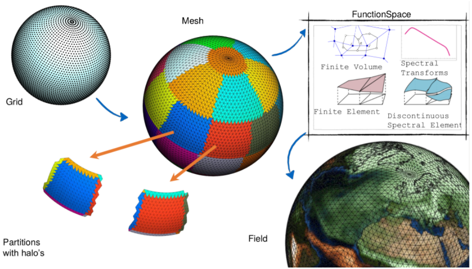
Flexible data structures for complex memory hierachies

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

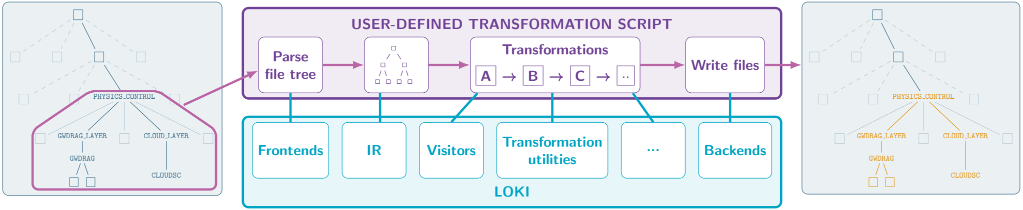
- Modern C++ library with Fortran interfaces [2]
- Data structures for numerical algorithms:
  - Remapping and interpolation
  - Gradient, divergence, laplacian
  - Spherical Harmonics transforms
  - **Increasing accelerator-awareness!**



[2] 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.

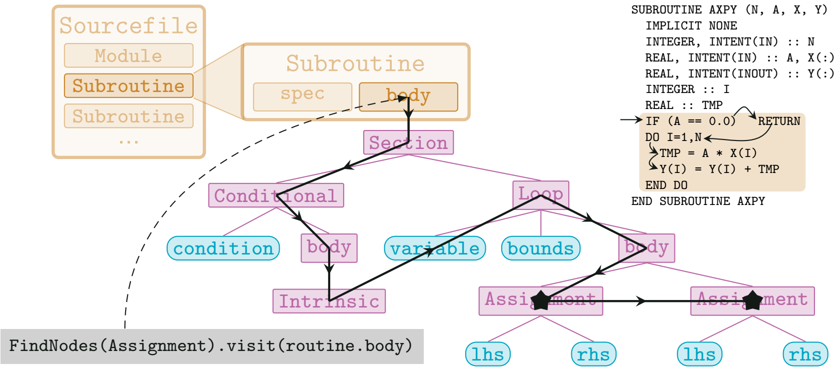
Source-to-source translation of physical parameterisations

- **Loki:** Programmable source-to-source translation tool written in Python
- Bulk transformation of source tree at compile time (“complex preprocessor”)
- Enable parallel development of science code and performance engineering
- Explore alternative programming model and device-specific code structures



Under the hood: Traversing expression trees

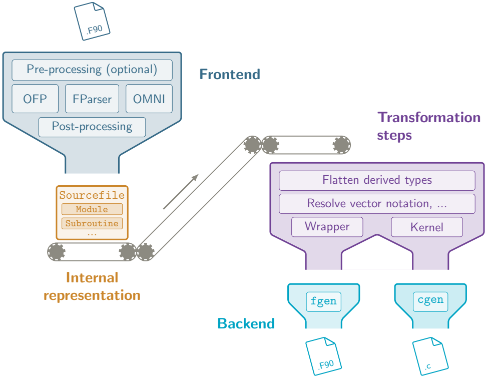
- Loki's internal representation (IR) is based on two levels of trees
  - **Custom** data structures for control flow (purple)
  - Enhanced **Pymbolic** nodes for expression trees (blue)



Loki – freely programmable source-to-source translation

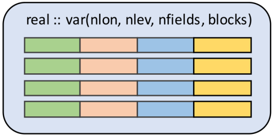
- Two-level internal representation: control flow and expression trees
- Visitors to search and transform code
- Growing library of utilities to aid encoding transformations

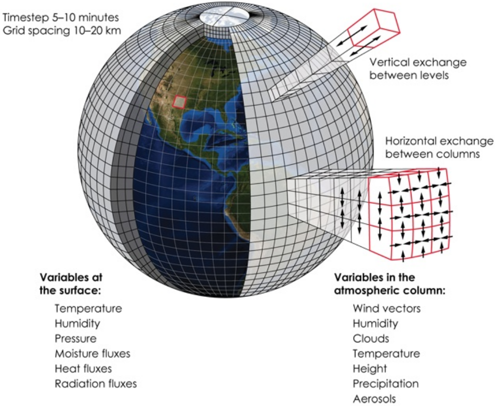
```
def remove_loops(self, routine, dimension):
    """
    Replace vector loop with it’s own body
    """

    map = {}
    for loop in FindNodes(Loop).visit(routine.body):
        if loop.variable == dimension:
            map[loop] = loop.body

    routine.body = Transformer(map).visit(routine.body)
```



Single-column layout in physical parameterisations

- No data dependencies between columns: Lots of parallelism! [3]
- Scientific kernel can be developed and tested for a single column
- Columns are stored in a block layout traversed with a high OpenMP loop
- Resulting memory layout packs sub-blocks of arrays into cache-friendly chunks (NPROMA)



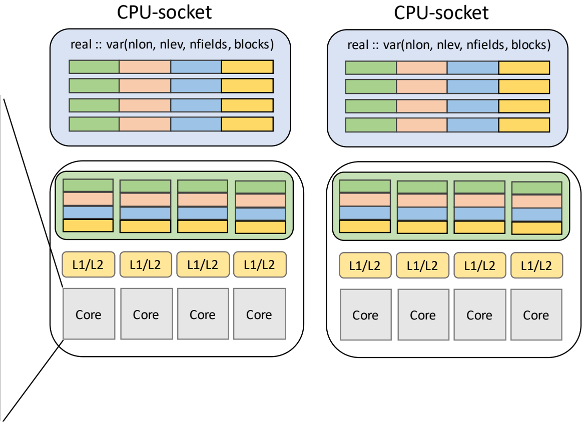


[3] Valentin Clement et al. "The CLAW DSL: Abstractions for Performance Portable Weather and Climate Models". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '18. 2018. ISBN: 9781450358910. DOI: 10.1145/3218176.3218226.

IFS: Memory data layout and parallelisation

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

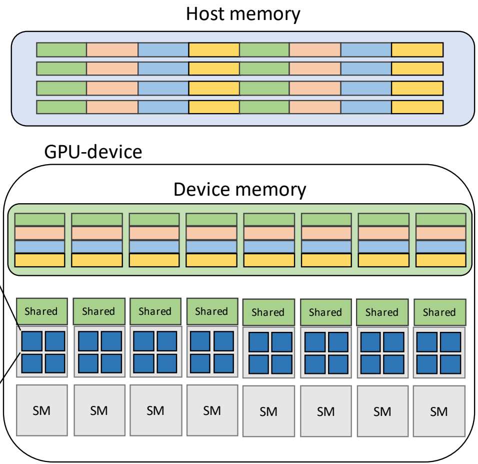
```
SUBROUTINE KERNEL(nlon,nlev,var1,var2,…)
  real :: var1(nlon, nlev)
  real :: var2(nlon)

  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
```



Automatically mapping memory-blocked CPU code to GPUs

CLOUDSC (/ecmwf-ifs/dwarf-p-cloudsc)

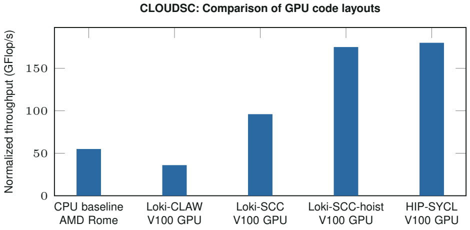
- Standalone version of cloud microphysics
- Representative parallelisation, memory layout
- Challenging to optimise (high register pressure)

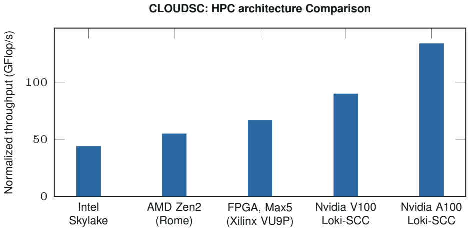
Evaluation of GPU code transformations

- Pure compute throughput comparison (no offload)
- Comparison of different Fortran GPU loop strategies
- HIP-SYCL: Manual translation from C variant

Per-chip/socket performance comparison

- Chip-to-chip compute thoughput comparison
- PCIe offload cost ignored for single kernel
- FPGA results are memory-bandwidth limited! [4]





[4] James Stanley Targett et al. “Systematically migrating an operational microphysics parameterisation to FPGA technology”. In: *29th IEEE FCCM 2021, Orlando, FL, USA, May 9-12, 2021*. IEEE, 2021, pp. 69–77. DOI: 10.1109/FCCM51124.2021.00016.

Preparing IFS for HPC accelerators via source-to-source translation

Hybrid 2024: Internal project to prepare IFS for HPC accelerators

- Develop GPU and accelerator capabilities alongside scientific development
- Incremental adaptation of IFS components to different accelerator types
- Use library APIs, GPU-enabled data structures and source-to-source translation to separate technical concerns from scientific user code

Loki: Freely programmable source-to-source translation

- In-house programmable source-to-source translation tool, under active development to encode (instead of commit) code changes
- Bulk transformation of source tree at compile time (“complex preprocessor”)
- Flexible and powerful, but does require explicit encoding of recipes and thus expertise

Thank you! Any questions?

michael.lange@ecmwf.int
MLange805