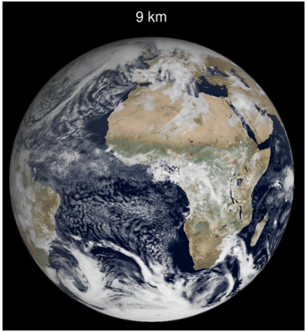
# Preparing the IFS for HPC accelerator architectures

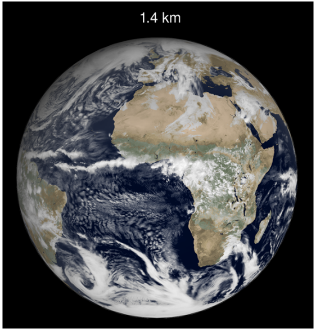
M.Lange

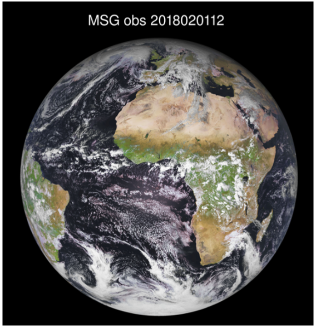
michael.lange@ecmwf.int

European Centre for Medium-Range Weather Forecasts

# Global Weather and Climate Simulations at 1km resolution



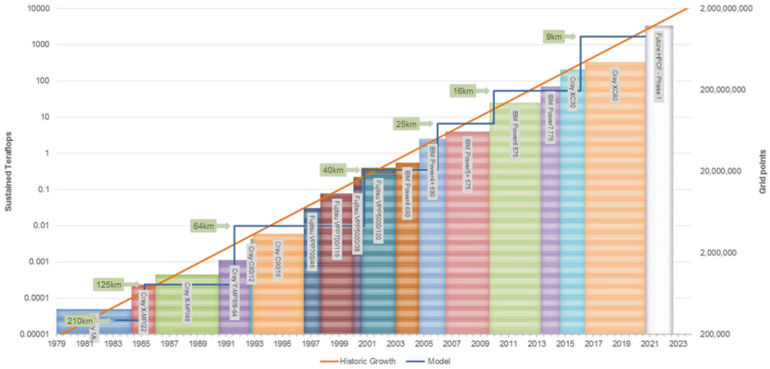




*"Albeit only a single realization due to its considerable computational cost, the resulting model output provides a reference and guidance for future simulations."* [1]

---

[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`.

# IFS - Sustained performance increase since 1979

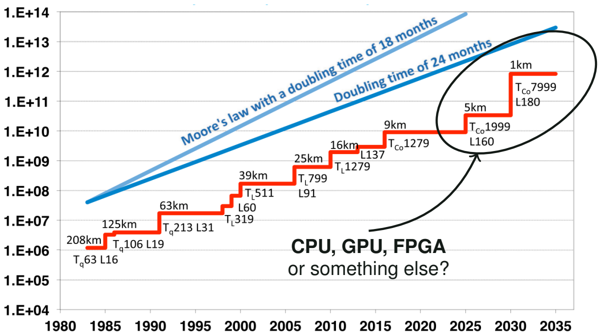



| Machine | Sustained/Peak % |
|---|---|
| Cray 1A | 31% |
| Cray X-MP/22 | 50% |
| Cray X-MP/48 | 48% |
| Cray Y-MP8/8-64 | 42% |
| Cray C90/12 | 39% |
| Cray C90/16 | 39% |
| Fujitsu VPP700/46 | 31% |
| Fujitsu VPP700/116 | 30% |
| Fujitsu VPP700E/48 | 31% |
| Fujitsu VPP5000/38 | 30% |
| Fujitsu VPP5000/100 | 30% |
| IBM Power4 690 | 6% |
| IBM Power4+ 690 | 8% |
| IBM Power5+ 575 | 11% |
| IBM Power6 575 | 8% |
| IBM Power7 775 | 5% |
| Cray XC30 | 6% |
| Cray XC40 | 4% |

# So, Moore's Law was fun...

**Ambitious target of 1km resolution at 1SYPD requires ∼250x improvement[2]**

- Moore's Law is stagnating and continued performance growth is not guaranteed
- Emerging architectures can provide continued growth in computational power
- Software and infrastructure changes are required alongside hardware upgrades




[2] 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.

# Preparing IFS for HPC accelerators

**Grand vision: Accelerator-enabled multi-architecture IFS**

- **Aim:** Port and optimise model components for different accelerators
- **Challenge:** IFS is heavily optimized for deep-cache CPU architectures
- **Approach:** Use dedicated build-modes to target different programming models
- **Goal:** Develop accelerator capabilities alongside scientific development

**Involvement and synergies with many European projects**

- **Destination Earth:** High-resolution “Digital Twins” using EuroHPC hardware
- **Center of Excellence** with Atos, Nvidia, Mellanox
- **Several European projects:** DEEP-SEA, EUPEX, ESIWACE-2, ESCAPE-2, ...

*Please go and see the many talks highlighting these collaborations.*

**Bridging the gap: Specialised build-modes and new programming models**

- Use library APIs to separate technical and scientific code
- Develop control flow flexibility for hybrid execution and offload modes
- Use source-to-source translation for kernel performance optimisations

# Separation of concerns: Separating API from implementation

**Accelerator-enabled data structures and libraries**

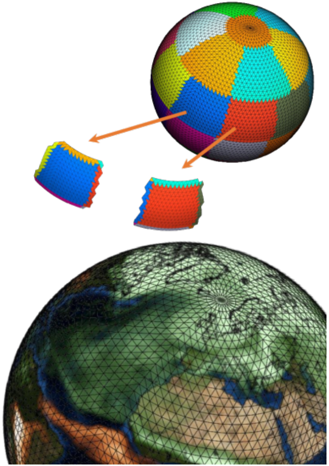
- Encoding technical detail behind clean library APIs
- Object-oriented data structures for increased flexibility
- Managing data placement in complex memory hierarchies

**Spectral Transforms: Accelerator-specific library backends**

- ESCAPE and INCITE program (*see talks on Friday!*)
- Develop accelerator-specific spectral transform as a library

**Atlas - A modern C++ data structure library**[3]

- Enable new numerical algorithm development
- Separate high-level concepts from implementation (grid, mesh, field vs. data storage, device memory)
- Accelerator-aware data structure and operators



[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.

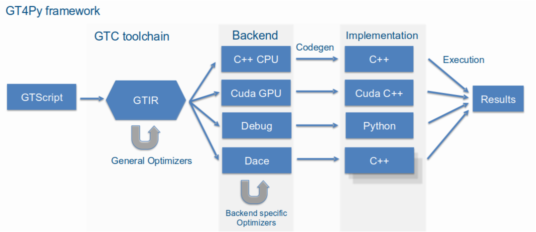
# High-level abstractions: IFS-FVM and the GridTools/GT4Py DSL

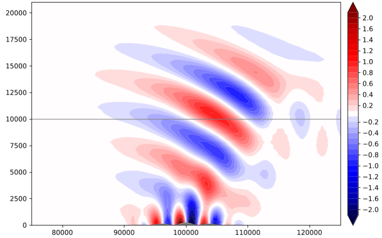
**Alternative dynamical core with different footprint**

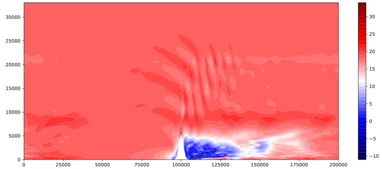
- Nonhydrostatic finite-volume formulation for IFS[4]
- Neighbour-only halo exchanges via Atlas fields

**DSL as route to accelerators and heterogeneous HPC**

- **FVM-LAM:** Structured grid 3D dy-core validated
- **Goal:** Global unstructured FVM in GridTools/GT4Py





[4] C. Kühnlein et al. "FVM 1.0: a nonhydrostatic finite-volume dynamical core for the IFS". In: *Geoscientific Model Development* 12.2 (2019), pp. 651–676. DOI: 10.5194/gmd-12-651-2019. URL: https://gmd.copernicus.org/articles/12/651/2019/.

# Source-to-source translation for gridpoint compute

### Loki: Programmable source-to-source translation[5]

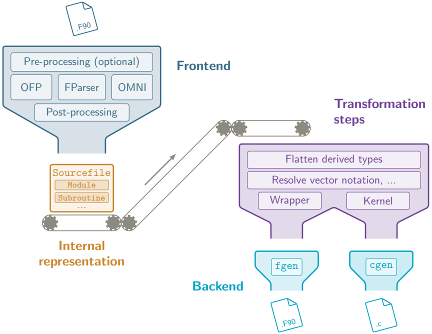
- Compiler technology: IR trees, visitors, ...
- Transformations are user-defined by experts

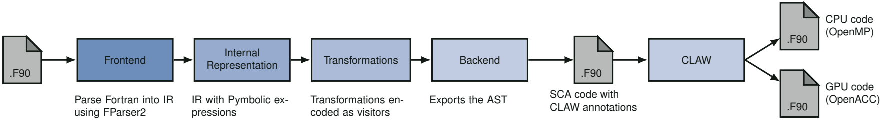
### Encode changes rather than commit them

- Bulk-transformation at compile time
- Can act as a complex pre-processor
- Can make transformations specific to IFS

### Explore alternative programming models

- Fortran@CPU to Fortran@OpenACC
- Feed downstream tools and DSLs



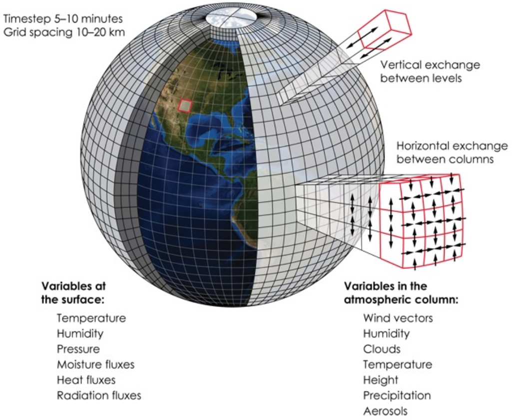




[5] B. Reuter et al. “Poster: Loki - A Source-to-Source Translation Tool for Numerical Weather Prediction Codes”. In: *PASC 2021, July 5-9*. 2021.

# IFS - Physical Parameterisations

Single column abstraction to generalise data parallelism

- "Physics" form a large part of the code base
- Physics have no clear performance profile and use a mixture of numerical methods
- Common pattern: **No data dependencies** between columns, so lots(!) of parallelism
- Scientific kernel can be **developed** and **tested** for a **single column**
- **NPROMA:** Columns are stored in a block layout with high OpenMP loop
- CLAW source-to-source compiler[6] can exploit **inherent parallelism** via **architecture-specific code generation**




[6] 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.

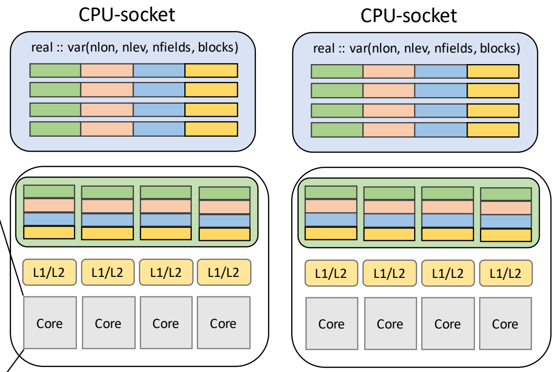
# IFS - Memory data layout and parallelisation

```
!$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
```



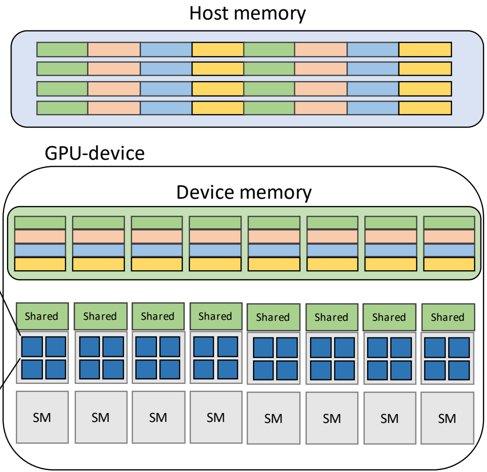
# IFS - Memory data layout and parallelisation

```
!$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 - ESCAPE dwarf**

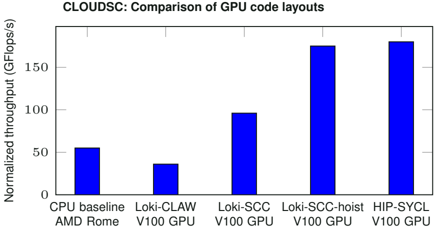
- Standalone version of cloud microphysics scheme
- Representative parallelisation and memory layout
- Optimisation is challenging (high register pressure)

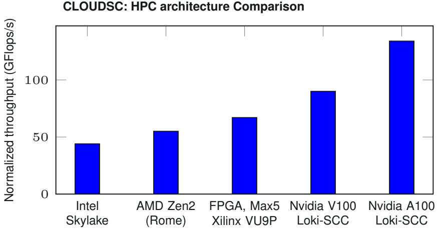
**Evaluation of different GPU code transformations**

- **Loki-CLAW:** GPU-parallel, no memory blocking
- **Loki-SCC:** GPU-parallel with memory blocking
- **Loki-SCC-hoist:** Hoisted temporary arrays
- **HIP-SYCL:** Manual translation from C variant

**Per-chip performance comparison**

- Chip-to-chip compute thoughput comparison
- PCIe offload cost ignored for single kernel
- FPGA results are memory-bandwidth limited![7]




[7] 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.

# Thank you! Any questions?