Accelerating extreme-scale Numerical Weather Prediction

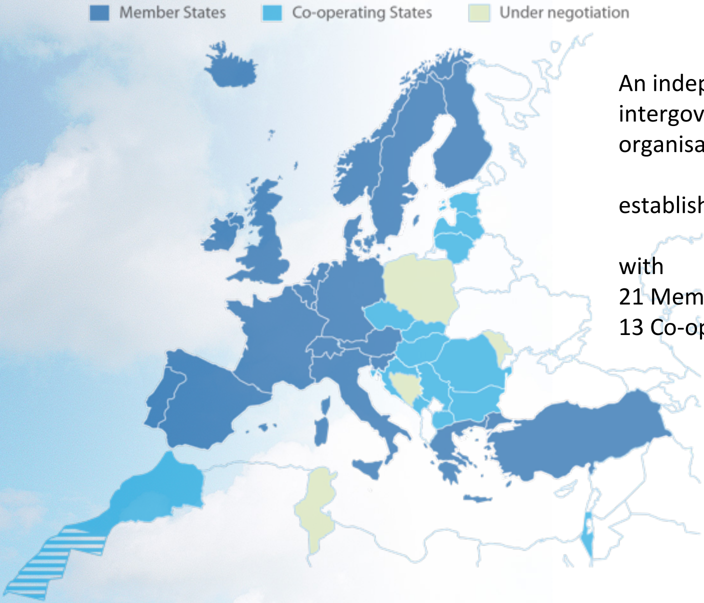
Willem Deconinck, Christian Kühnlein, George Mozdzynski, Piotr K. Smolarkiewicz, Mats Hamrud, Nils P. Wedi

ECMWF
European Centre for Medium-Range Weather Forecasts

PPAM
September 2015, Krakow

willem.deconinck@ecmwf.int

ECMWF



An independent intergovernmental organisation

established in 1975

with
21 Member States
13 Co-operating States

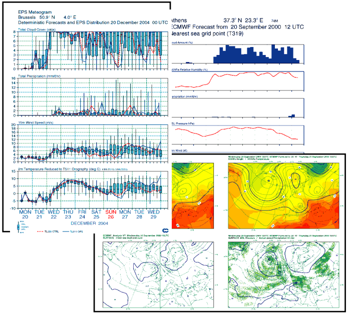
Who are we and what do we do?

Weather Forecasts	We produce global weather forecasts
Medium-Range	Up to 15 days ahead. Our products also include monthly and seasonal forecasts and we collect and store meteorological data.

Reading, United Kingdom

What do we have to achieve this?

People	About 260 staff, specialists and contractors
Equipment	State-of-the-art supercomputers and data handling systems
Budget	£50 million per year

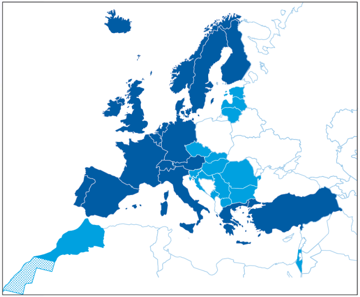


Global observation system

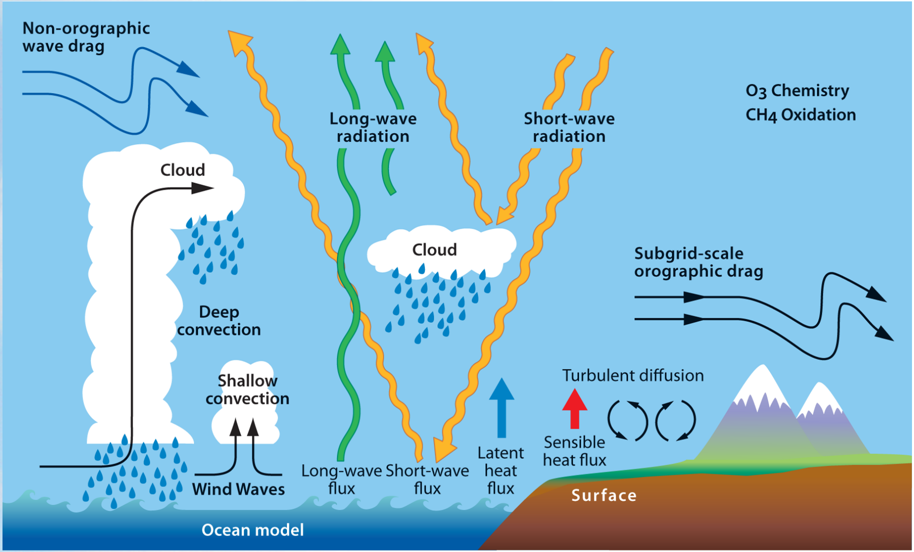
Global numerical weather forecasts

Users

National weather services

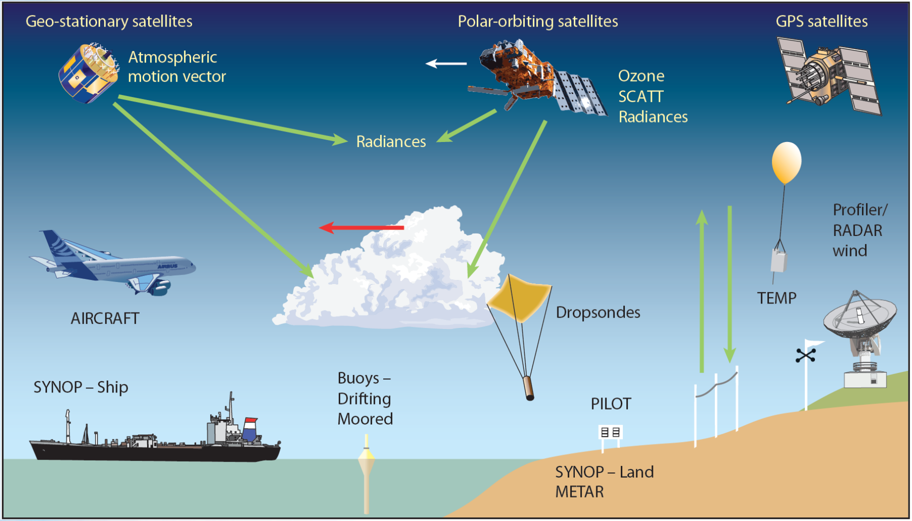


Physical processes



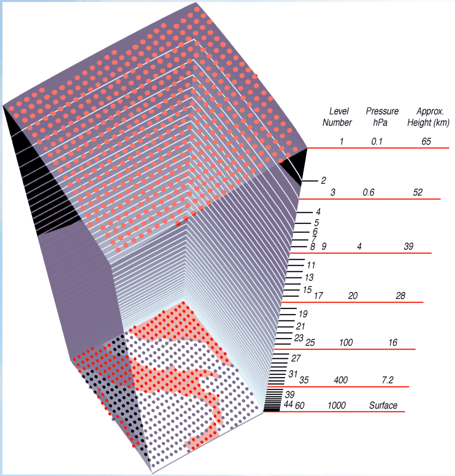
ECMWF

Observations used to assimilate new initial condition



ECMWF

Integrated Forecasting System (IFS)

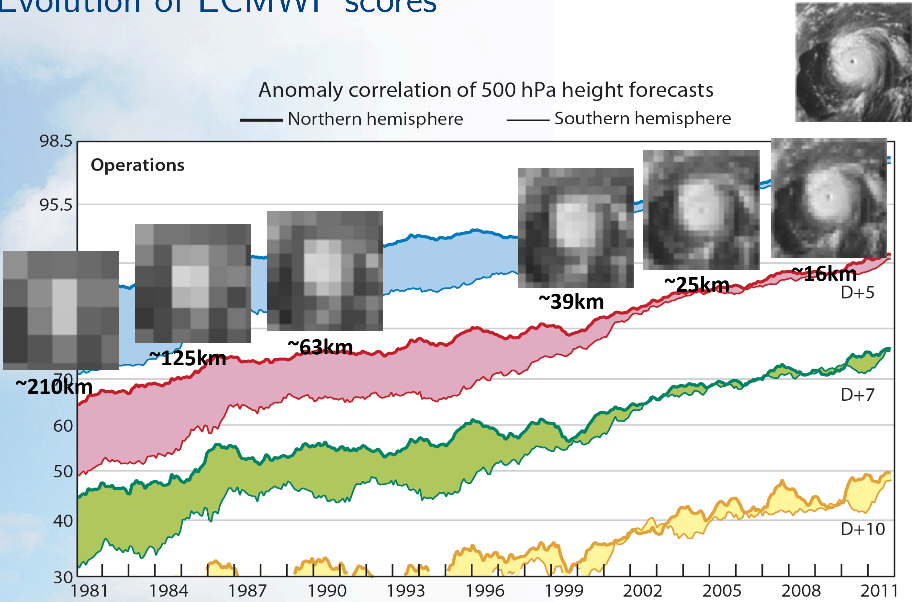


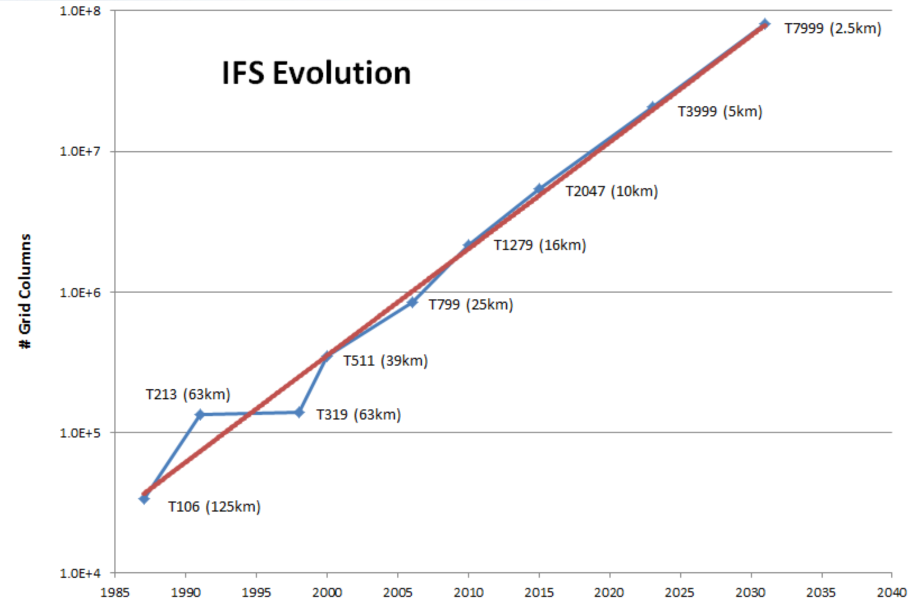
- Medium-Range ~10 days ahead
- IFS: Global NWP model
- 1279 spectral wave lengths ~16km horizontal resolution
- 137 levels vertically
- Run model in less than 1 hour (240 times faster than real-time)
- 51 ensemble forecasts for confidence interval

- double resolution every 8 years
- end this year: ~9km horizontal resolution
- increasing demand on computational power

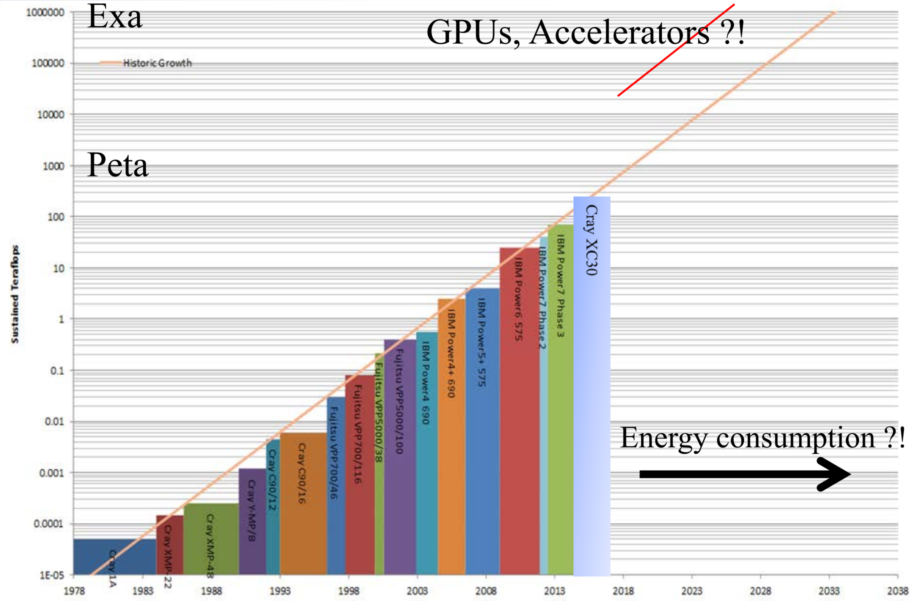
- 2 Cray XC30 systems for redundancy
- 3500 dual-socket compute nodes per system
- 19 Cray XC30 cabinets equipped with Intel Ivy Bridge processors
- 3,593 teraflops peak performance per system

Evolution of ECMWF scores



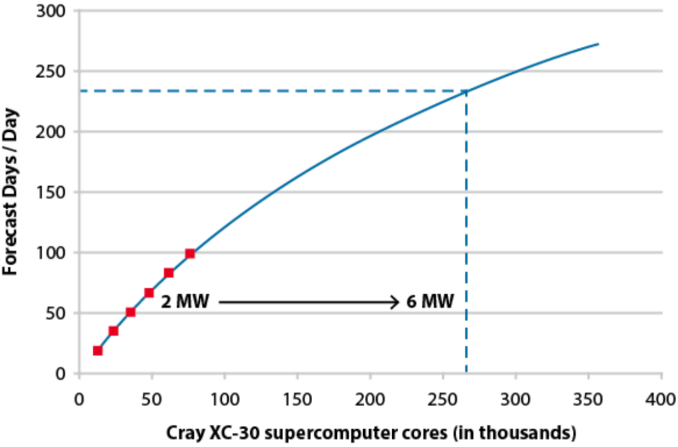

IFS Evolution
Grid Columns
1.0E+8
1.0E+7
1.0E+6
1.0E+5
1.0E+4
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
T7999 (2.5km)
T3999 (5km)
T2047 (10km)
T1279 (16km)
T799 (25km)
T511 (39km)
T319 (63km)
T213 (63km)
T106 (125km)

Still feasible to obtain Exaflop?



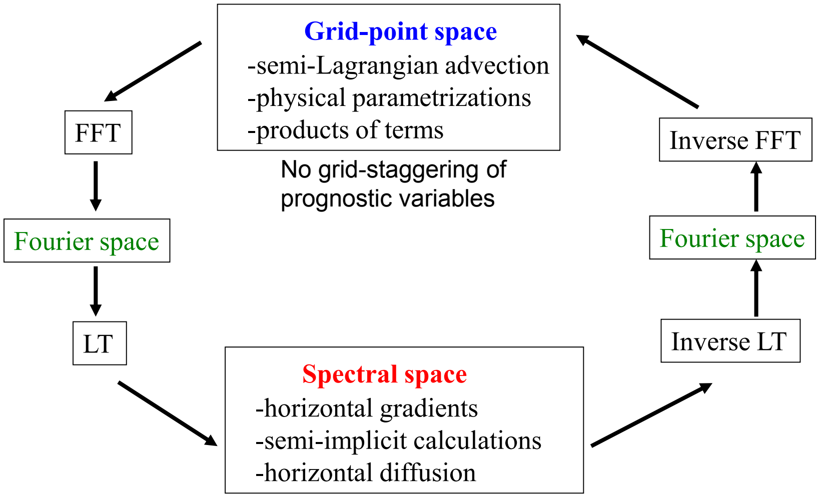
How does IFS scale?

2.5km L137 performance



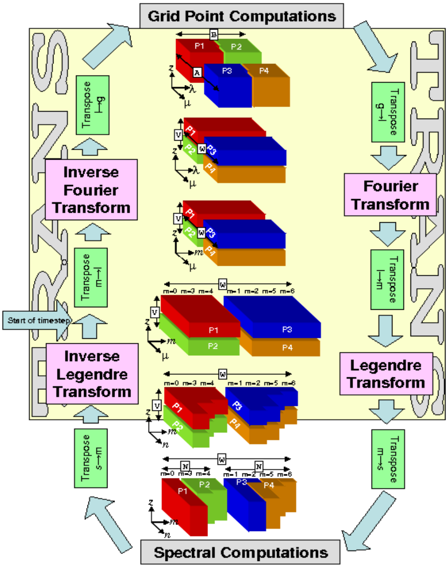
Operational target: 240 Forecast Days/Day → 270k cores!!! 6MW!!!
For ONE forecast... System needs to be ∼20 times that size... 120MW?

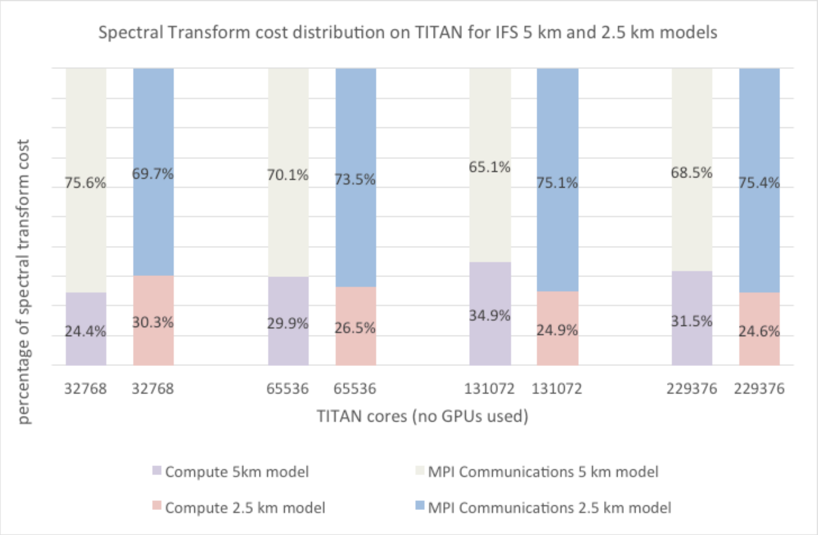
IFS: Spectral Transform Model



FFT: Fast Fourier Transform, LT: Legendre Transform

Several transpositions within the spectral transforms need to communicate, e.g. using MPI_AlltoAllv




Spectral Transform cost distribution on TITAN for IFS 5 km and 2.5 km models
percentage of spectral transform cost
75.6%
69.7%
70.1%
73.5%
65.1%
75.1%
68.5%
75.4%
24.4%
30.3%
29.9%
26.5%
34.9%
24.9%
31.5%
24.6%
32768
32768
65536
65536
131072
131072
229376
229376
TITAN cores (no GPUs used)
Compute 5km model
MPI Communications 5 km model
Compute 2.5 km model
MPI Communications 2.5 km model

A few shortcomings of IFS

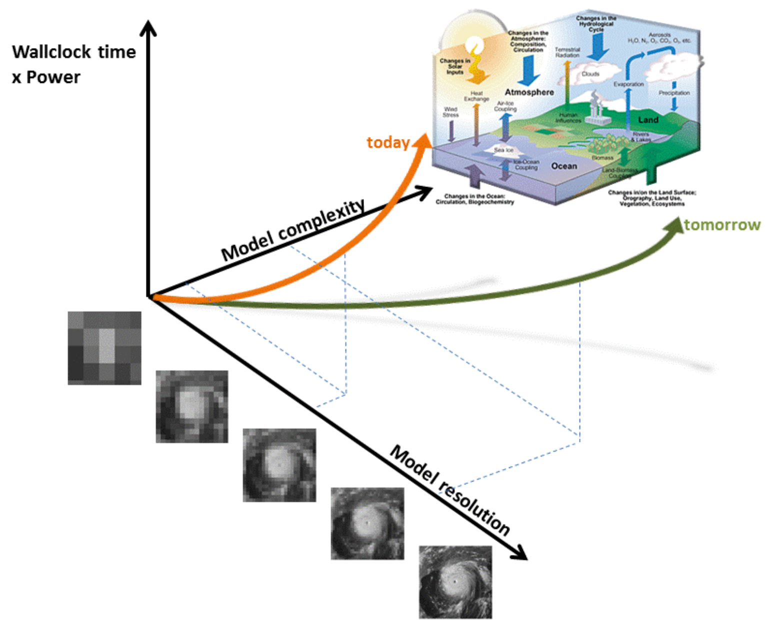
- Gradient computations are expensive (but very accurate)
    1. Direct transform of variable
    2. Take derivative in spectral space
    3. Inverse transform of derivatives

  Involves global communication
- 30 yrs old, 2 million lines, Fortran90, co-developed with Météo France
    - Even simple refactoring takes years (OOPS project)
    - Cyclic dependencies
    - No abstractions
    - IFS... IF (METEOFRANCE) THEN ... ELSEIF (ECMWF) THEN ...

So... *how* will IFS ever be able to cope in a ever-changing world?

- “Free lunch is over” (Herb Sutter)
- Evolving (or rather radically different) parallel computing paradigms
- Which hardware will offer the best performance at minimal cost?

Increasing model complexity and resolution



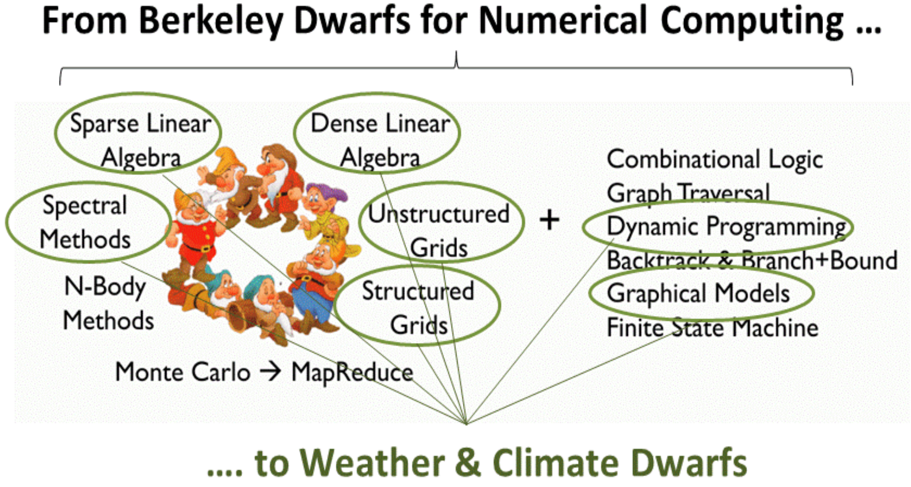
ESCAPE (H2020 project)

Energy-efficient Scalable Algorithms for Weather Prediction at Exascale

- ECMWF (host)
- Regional NWP centres (BE,FR,DK,CH,DE)
- Universities (UK)
- HPC centres (IE,PL)
- Hardware vendors (NVIDIA,Bull,Optalysys)
- 3 years
- Starts 01-10-2015 (in 3 weeks)

ESCAPE will address the question how specialised compute units and accelerator technologies may be used that offer unprecedented exascale, energy-efficient floating point rates in the future.

Identify "Weather & Climate Dwarfs" (in analogy to Berkeley Dwarfs)

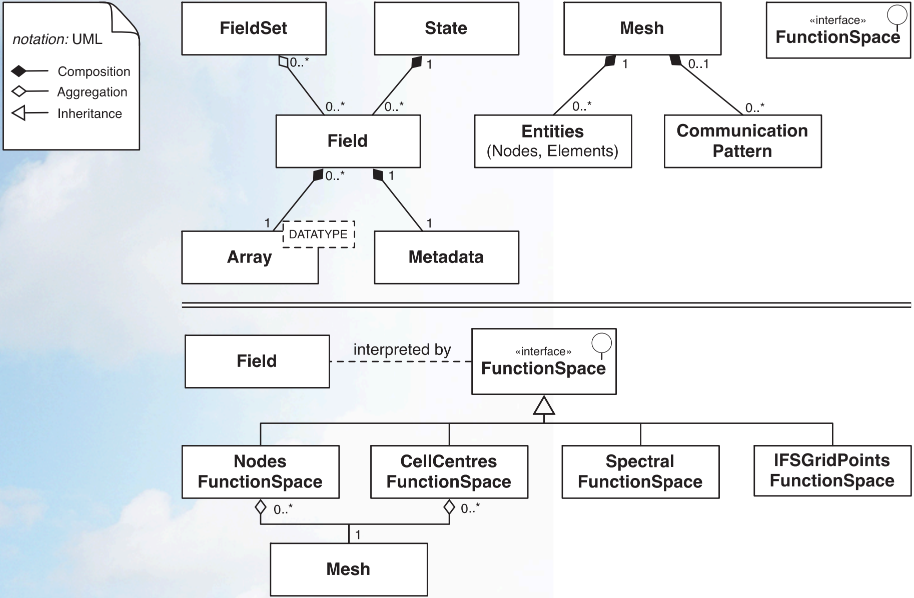


Atlas, a parallel flexible datastructure framework

In Greek mythology, Atlas was the primordial Titan who held up the sky.
He was the son of Aether (personification of Air) and Gaia (personification of Earth).

- C++ code with Fortran 2003 OO-interface
- Hierarchical composition of data: *Mesh, State, FieldSet, Field, FunctionSpace, Metadata*
- Unstructured distributed meshes
- Mesh-generation – Domain Decomposition – Halo-creation
- Communication Patterns (HaloExchange, Gather, Scatter)
- Collection of Interpolation, Gradient, . . . algorithms (not yet)
- Collection of Input/Output formats (grib,netcdf,gmsh)

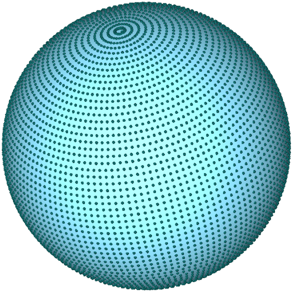
Atlas components



Mesh generation

Reduced Gaussian Grid

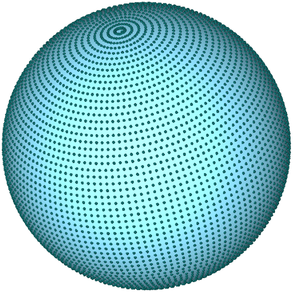
- Uniform distribution along longitudes
- Gaussian distribution along lattitudes
- Progressively reduce number points towards poles



Mesh generation

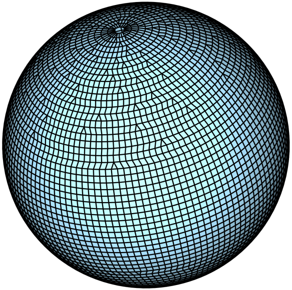
Reduced Gaussian Grid

- Uniform distribution along longitudes
- Gaussian distribution along lattitudes
- Progressively reduce number points towards poles



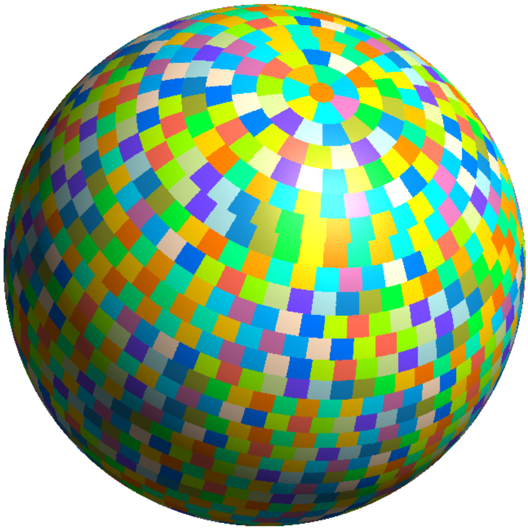
Reduced Gaussian Mesh

- New element-based schemes can complement / replace IFS spectral transform schemes
  - Local gradient computations
  - Conservative transport



Parallelisation

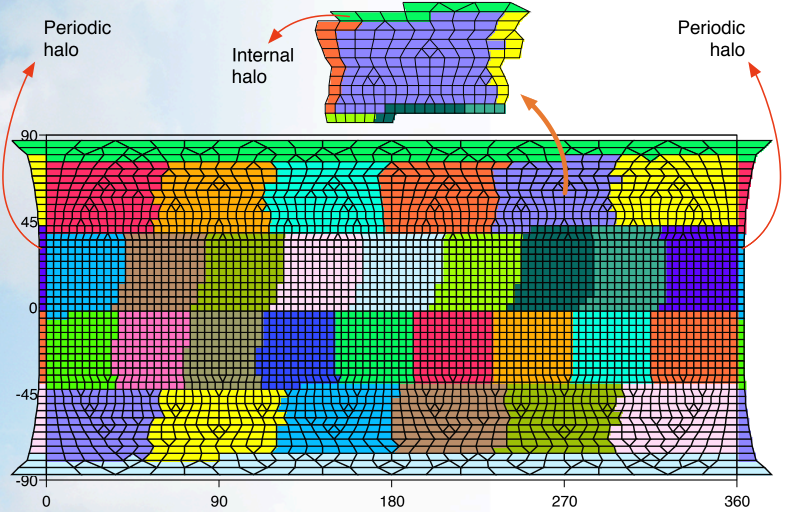
Equal Regions Domain Decomposition



1600 partitions
~9km resolution
~6.6 million nodes

Parallelisation

Equal Regions Domain Decomposition



Field

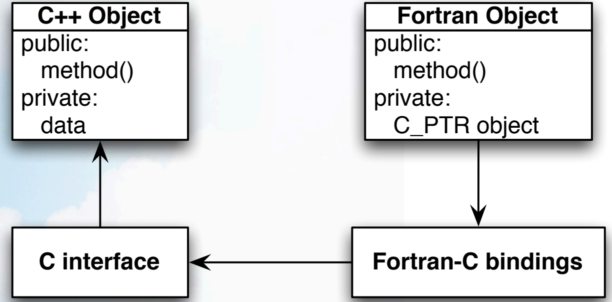
```
// Create wind field
Field windfield( "wind",
                 DataType<double>(),
                 make_shape(nb_nodes,nb_levels,3) );
windfield.metadata().set("units","m/s");

ASSERT( windfield.size() == nb_nodes*nb_levels*3);
ASSERT( windfield.name() == "wind");
ASSERT( windfield.rank() == 3);
ASSERT( windfield.datatype().str() == "real64");

// Access to data
ArrayView<double,3> wind ( windfield );
for( size_t jnode=0; jnode<nb_nodes; ++jnode ) {
  for( size_t jlev=0; jlev<nb_levels; ++jlev ) {
    wind(jnode,jlev,0) = ... /* X-component */
    wind(jnode,jlev,1) = ... /* Y-component */
    wind(jnode,jlev,2) = ... /* Z-component */
  }
}
```

All scientific codes at ECMWF are written in Fortran

- Keep Fortran for numerical algorithms
- C++ for . . .
  - All data memory management
  - Message passing

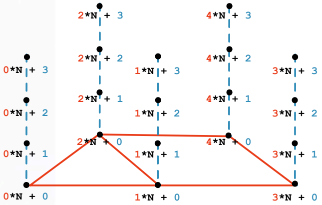
Expose the flexible and dynamic data-structure to Fortran

Create for every C++ class, a Fortran Derived Type



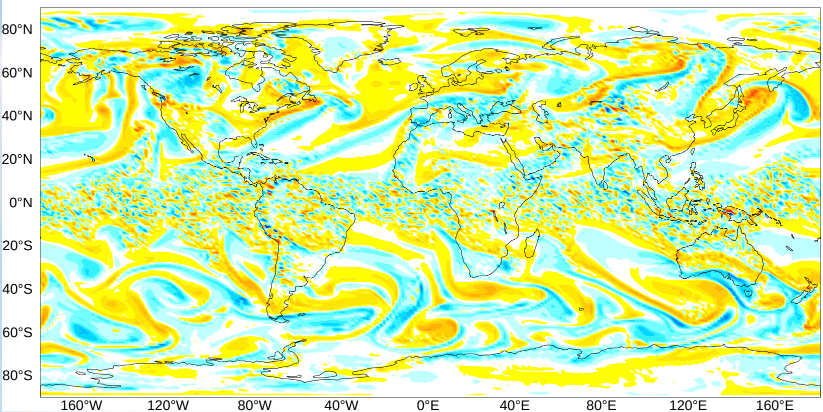
```fortran
type(atlas_Field) :: windfield
real(8), pointer :: wind(:,:,:)

! Create field
windfield = atlas_Field( "wind", &
  &                      atlas_real(8), &
  &                      (/3,nb_levels,nb_nodes/) )

! Access data
call windfield%data(wind)
do jnode=1,nb_nodes
  do jlev=1,nb_levels
    wind(1,jlev,jnode) = ... ! X-component
    wind(2,jlev,jnode) = ... ! Y-component
    wind(3,jlev,jnode) = ... ! Z-component
  enddo
enddo
```

Contiguous 1-dimensional array mapped to rank 3 Fortran array



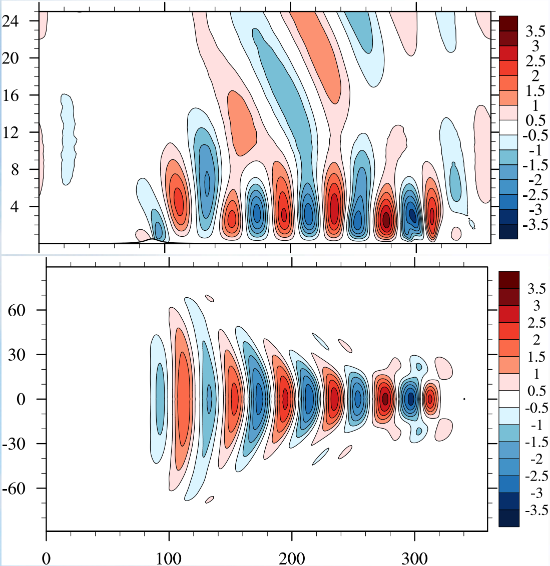
Developing new model built on Atlas

"A hybrid all-scale finite-volume module for stratified flows on a rotating sphere", Smolarkiewicz et al. (In review with JCP)



Unstructured horizontally, structured vertically, MPI + OpenMP

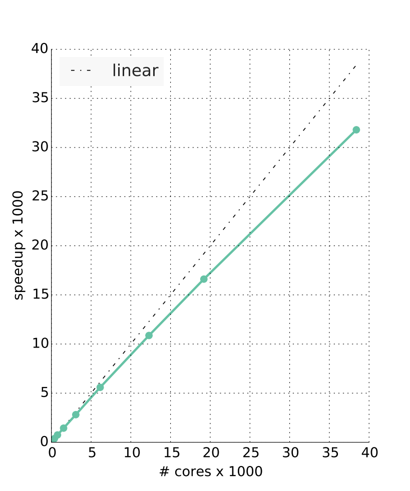
Quasi-2D Sheared Orographic Flow on Small Planet



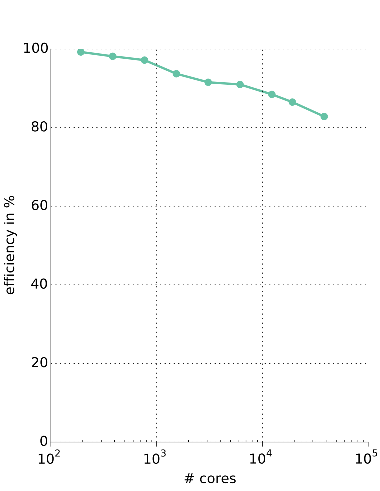
- ~0.7° resolution
- Non-Hydrostatic gravity wave response
- vertical velocity
- top: XZ slice at equator
- bottom: XY slice at 3km height

Strong Scaling results

speedup



efficiency



No attempts have yet been made to optimize efficiency

Conclusions

- ESCAPE will address the question how exa-scale could be achieved for Weather & Climate applications
- Atlas datastructure framework as a basis for collaborating with partners
- . . . and allow research in alternative numerical algorithms previously not possible with 30 year old legacy code
- Fortran interfaces to C++ design can modernize legacy code
- New model in development based on unstructured edge-based finite volume code

Thank you for your attention!

Contact:

willem.deconinck@ecmwf.int