Building a Performance Portable Software System for the Met Office's Weather and Climate Model, LFRic

S.V. Adams[a], M. Ashworth[b], R.W Ford[c], M. Hambley[a], J.M. Hobson[a], I. Kavcic[a], C.M. Maynard[a,d], T.Melvin[a], E.H. Muller[e], S. Mullerworth[a], A.R. Porter[c], M. Rezny[f], G.Riley[b], B.J. Shipway[a], S.Siso[c], R. Wong[a]
[a] Met Office, [b] University of Manchester, [C] STFC Hartree Centre, [d] University of Reading, [e] University of Bath, [f] Monash University

LFRic: Scalability and flexibility on future HPCs

LFRic [1] is the weather and climate model being developed by the Met Office to replace the Unified Model (UM) for exascale computing architectures. A domain specific language (DSL) has been developed to enable single science source code which can then be run on different architectures. It exploits a domain specific compiler and code generator called PSyclone to generate the different parallel code for different architectures. The general concepts of the DSL are illustrated and scaling results for MPI, MPI and OpenMP on the Met Office XC40 machine are shown. Preliminary experiments with OpenACC code for GPUs are also shown for key computational kernels. Finally the development path for harnessing FPGA acceleration is described, showing the outcome of preliminary experiments on the architecture of the EuroExa project. The software system allows for single source science code development while at the same time allowing for different architecture targets and optimisations.

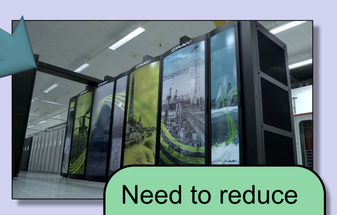
Diverse Future HPC: Accelerators, GPUs, ARM etc.

Exascale

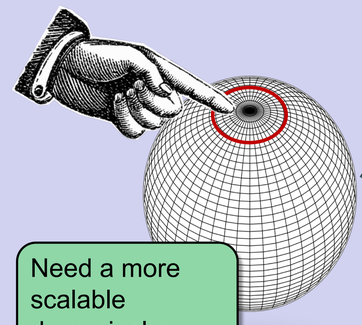
Scalability Problems

GungHo Project 工合

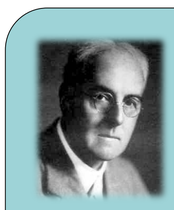
Met Office | NERC Science of the Environment | Science & Technology Facilities Council



Need to reduce cost and effort of porting codes from machine to machine.



Need a more scalable dynamical core.



Named after **Lewis Fry Richardson**
1922: *Weather Prediction by Numerical Process*

LFRic

Science code doesn't need to be changed for different HPC architectures

GHASP

PSyKAl Infrastructure: Parallel Systems, Kernels, Algorithms

Algorithm layer

Science code written in a Fortran-like* domain-specific language (DSL).

```
subroutine iterate_alg(rho,theta, u, … )
…loops, if-blocks etc…
  call invoke(
   pressure_grad_kernel_type(result,rho,theta),
   energy_grad_kernel_type  (result,rho,coords)
  )
…more invoke calls…
end subroutine
```

Hand-written

- Code is aligned with the written equations.
- **invoke** calls reference kernels that do the work.
- All operations are on whole fields.
- No references to optimisation.

Input

Algorithm code

Code generated from the DSL. The **invoke** calls are replaced with calls into the generated PSy code, one per invoke.

```
call invoke_1(result, rho, theta, coords)
```

Generated Fortran

* So like Fortran it would actually compile

Fortran call

Parallel-Systems (PSy) layer

Aims to optimise for different hardware

Optimisation script
Python

Input

PSyclone code generator

Generates

Generates

PSy-layer code

- Subroutines generated for each **invoke**.
- Access field data from the shared memory domain (e.g. MPI).
- Field data is broken down into chunks of one vertical column of atmospheric data.
- Kernels called once per column in parallel (e.g. OpenMP, OpenACC).

Generated Fortran

Fortran call

Kernel code

```
module pressure_grad_kernel_mod
```

Metadata tells PSyclone how to unpack data:

```
 type(arg_type) :: meta_args(3) = (/       &
    arg_type(GH_FIELD,  GH_INC,  W2),      &
    arg_type(GH_FIELD,  GH_READ, W3),      &
    arg_type(GH_FIELD,  GH_READ, W0)       &
    /)
 type(func_type) :: meta_funcs(3) = (/     &
    func_type(W2, GH_BASIS, GH_DIFF_BASIS),&
    func_type(W3, GH_BASIS),               &
    func_type(W0, GH_BASIS, GH_DIFF_BASIS) &
    /)
  integer :: iterates_over = CELLS
end type
```

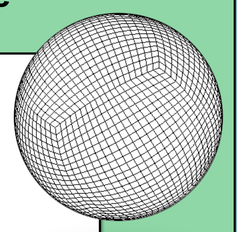
Science code (for a column of nlayers levels):

```
subroutine pressure_gradient_code( … )
! There is one cell of data on each level
do k = 1, nlayers
  ! Each cell has 1 or more data points
  do df = 1, num_dofs_per_cell
    result(map(df)+k)=theta(map(df)+k) * …
  end do
end do
end subroutine

end module
```

Hand-written Fortran

GungHo Dynamical Core



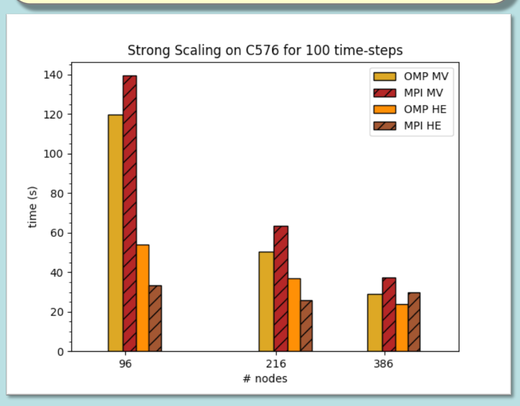
- New unstructured mesh:
  - no singularities at poles,
  - currently cubed-sphere.
- Indirect memory access for horizontal neighbours.
- Data for vertically adjacent cells is contiguous in memory.
- Finite-element formulation
  - coded to support high order.

Physics Parameterisations

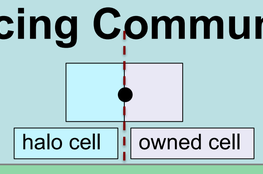
- Re-use of some UM code.
- Allows running the same physics code in both UM and LFRic.
- Finite-difference parameterisation schemes coupled to the finite element dynamical core.

Reducing Communication

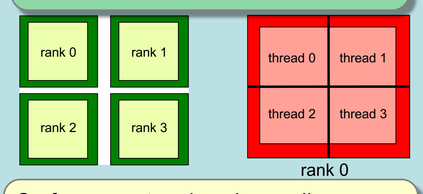
PSyclone can generate OpenMP as a transformation.
Further optimizations can reduce communication *e.g.* Redundant computation into halo.



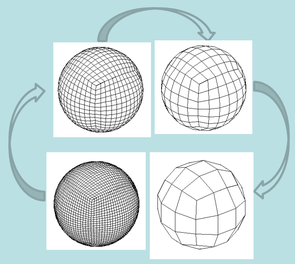
Time spent in the matrix-vector (MV) and halo-exchange (HE) on Met Office Cray XC40, using Intel 17 compiler. MPI denotes 36 MPI ranks per node, OMP 6 MPI ranks per node, 6 OMP threads per rank.



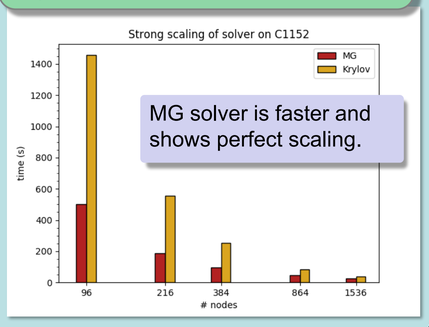
Data points shared with halo-cells are repeated in two partitions. Updates can require computation into halos. They can either be computed on one partition and sent with a halo exchange, or redundantly computed on all partitions to avoid halo comms. On a C576 cubed sphere (equiv. to ~17 Km), 96 nodes the number of halo exchanges was reduced 7.5x using redundant computation.



Surface area to edge size scaling implies OMP code has less redundant halo to compute.



Replacing the Krylov subspace solver (BiCGstab) for the Helmholtz equation (pressure) with a Multi Grid (MG) solver has reduces the number of global sums required.

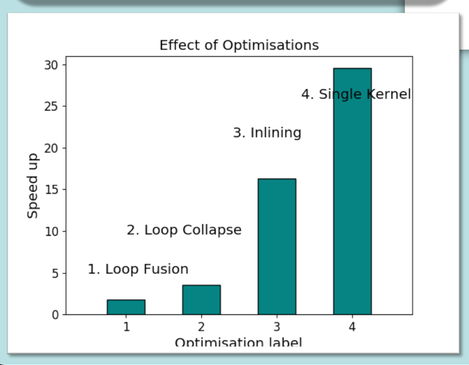


MG solver is faster and shows perfect scaling.

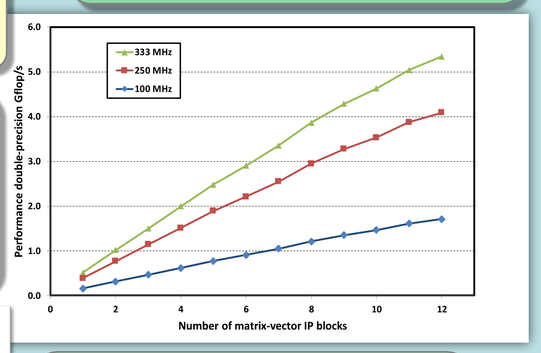
LFRic Microbenchmarks

EuroExa project: ARM CPUs + FPGA accelerator prototype → low power system
LFRic one of several applications.

Ported using High-level *Synthesis* tool from Xilinx Vivado.
Graph shows scaling versus IP block and clock speed.
Max 5.3 GFlop/s in double precision.
Comparable to CPU and GPU.
Significant benefits considering power.



PSyclone Kernel Extractor – Matrix-vector kernel.
Target optimisations / programming models different architectures.



Matrix-vector kernel parallelized with OpenACC – by hand.
Run on P9+volta GPU with PGI compiler.
OpenACC in PSy layer similar to openMP, but with data region.
Use OpenACC vector across cells in column in the kernel.
Optimisations: Loop order and directives
Alan Gray from NVIDIA [2].

Met Office FitzRoy Road, Exeter, Devon, EX1 3PB United Kingdom
Tel: +44 1392 885038 Email: christopher.maynard@metoffice.gov.uk

[1] S.V. Adams, *et al*. *LFRic: Meeting the challenges of scalability and performance portability in Weather and Climate models, Journal of Parallel & Distributed Computing,* 2019 https://doi.org/10.1016/j.jpdc.2019.02.007
[2] A. Gray. *Optimizing an OpenACC weather simulation kernel* https://www.openacc.org/blog/optimizing-openacc-weather-simulation-kernel

The FPGA work was carried out with support from the EuroExa project (Grant Agreement 754337), funded by the European Union Horizon 2020 Research and Innovation Programme