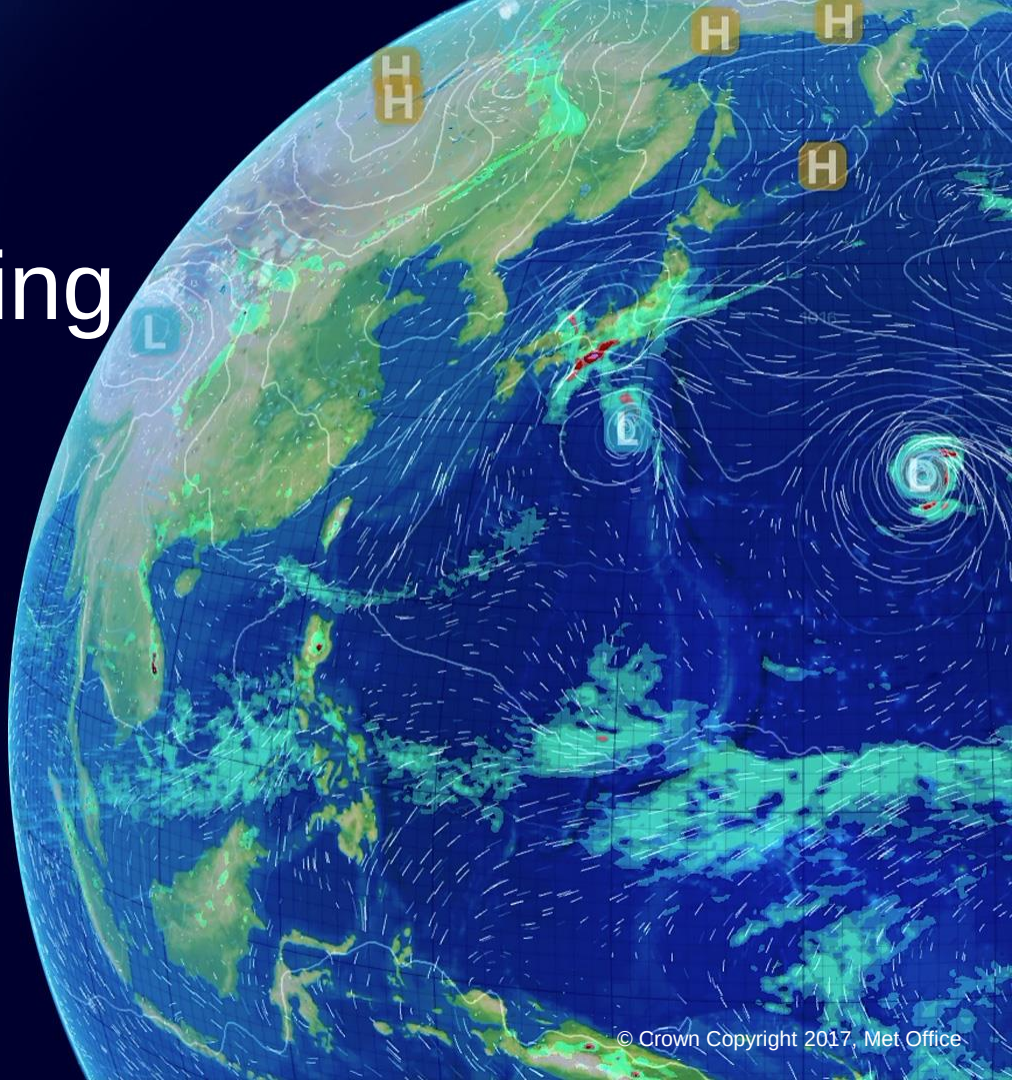
In-Situ Post-Processing with XIOS in LFRic

Samantha V. Adams

SIG-IO-UK

6th March 2019, Reading University, UK.

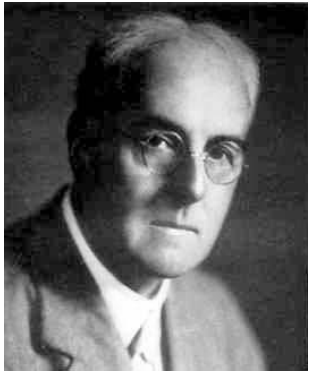


Talk Overview

- Recap: Background and Motivation for the LFRic project
- Post-Processing in the Met Office
- XIOS capabilities
- Recent performance results
- Next Steps

Met Office

What is LFRic?

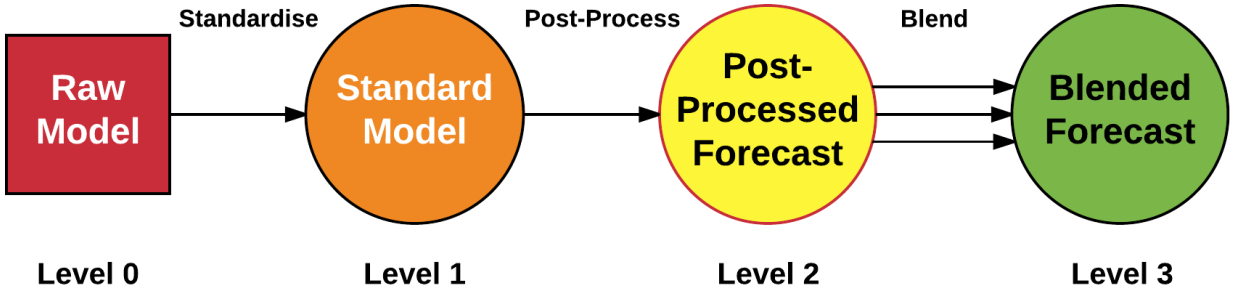


- **L**ewis **F**ry **Ric**hardson
- A project to rewrite Met Office modelling infrastructure
- GungHo Project Recommendations (*Met Office, NERC, STFC*)
- Develop science for a new dynamical core
  - Keep the best of current MO dynamical core (EndGame)
  - Improve where possible (e.g. Conservation)
- LFRic infrastructure will address two main (computational science) issues:
  - Scalability looking forward to Exascale
  - Flexible deployment for future HPC architectures

IMPROVER – a new post-processing workflow for the Met Office

- **Drivers for a new system:**
  - Science: exploitation of new capability
  - Programme: facilitate transformation
  - Standards: data and technology
- **Timescales**
  - Alpha release March 2019
  - Initial Operational Capability in 2020
  - Full migration to new system in 2021

IMPROVER – a new post-processing workflow for the Met Office



Data Processing Levels to help categorise and govern the data

Level 0: **Raw** Numerical Weather Prediction (NWP) model forecast data

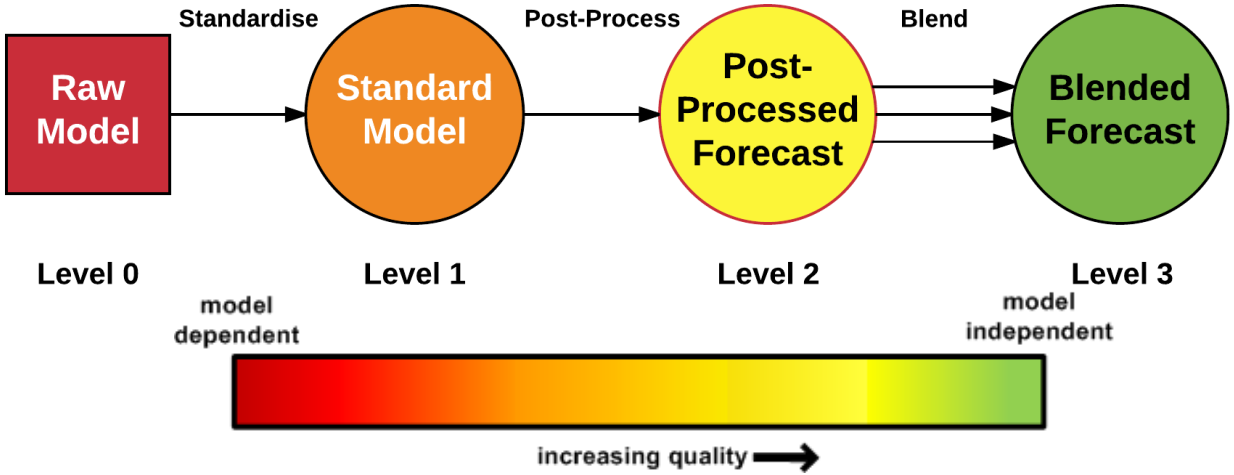
Level 1: NWP model data that has been '**standardised**'

Level 2: NWP model data that has had **post-processing** steps applied to it

Level 3: Post-processed NWP model forecasts from multiple models **blended** together to form a single best forecast, usually expressed primarily as a set of probabilities

Slide courtesy of Bruce Wright

IMPROVER – a new post-processing workflow for the Met Office



This represents a progression in:

- **Application of Scientific Correction**
- **Data Standardisation (decoupling from models)**

Slide courtesy of Bruce Wright

Met Office

XIOS Capability

- For our needs seems the best of available parallel I/O frameworks
- Client/Server architecture supporting parallel asynchronous I/O
- Supports parallel read and write of NetCDF and UGRID-NetCDF
- Proven on jobs ~10K cores in weather and climate domain
- Works with OASIS coupler now and coupling functionality is being added
- **Offers parallel ”in situ” post-processing via a workflow defined in the XML file(s)**
- **Offers the opportunity to go direct to Level 1 data**

Previous Performance Results (2017)

Scaling

- Run out to 14k cores with little/no I/O penalty. (but nowhere near operational resolution or science)
- Tuning I/O servers. As expected generally more == better. Reduces client wait time and no increase in overall run time…..BUT…

Impacts of a Lustre file system?

- With appropriate striping, can achieve low client wait time with fewer I/O servers

Diagnostic output loading

- With each 100 field (~112Gb) increase, approx +5% I/O penalty

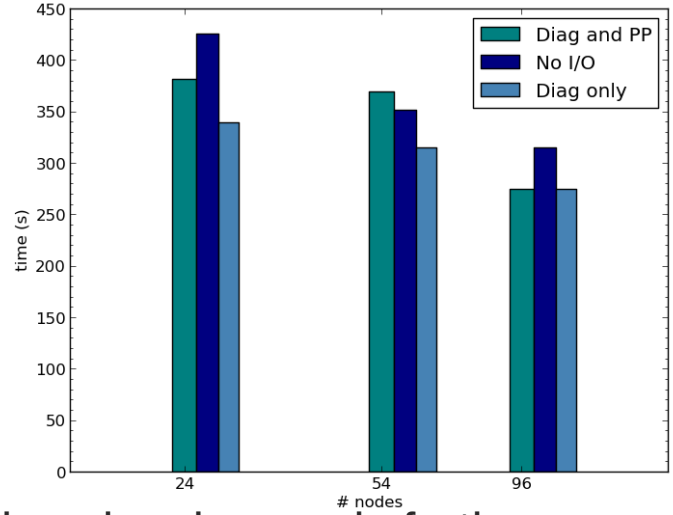
Latest Results

- Focus on Time-Meaning and Regridding
  - Use cases for upcoming Aquaplanet science trials
  - Regridding ability could impact how downstream post-processing is handled in the future and has implications for verification
- Assess impact on compute performance in comparison to non-I/O and regular diagnostic I/O scenarios
- Run time and memory usage

Job Configuration

- Standard baroclinic wave test
- Run is for **120 timesteps**, with **diagnostic frequency 20 timesteps**. (This makes for 6 writes over the course of the run)
- Executables were built using the LFRic Intel Fortran environment (**17.0.0.098**) including the latest PSyclone **1.7.0** and XIOS at **r1537** of the XIOS trunk
- All jobs run on Met Office XCS Cray XC40 machine

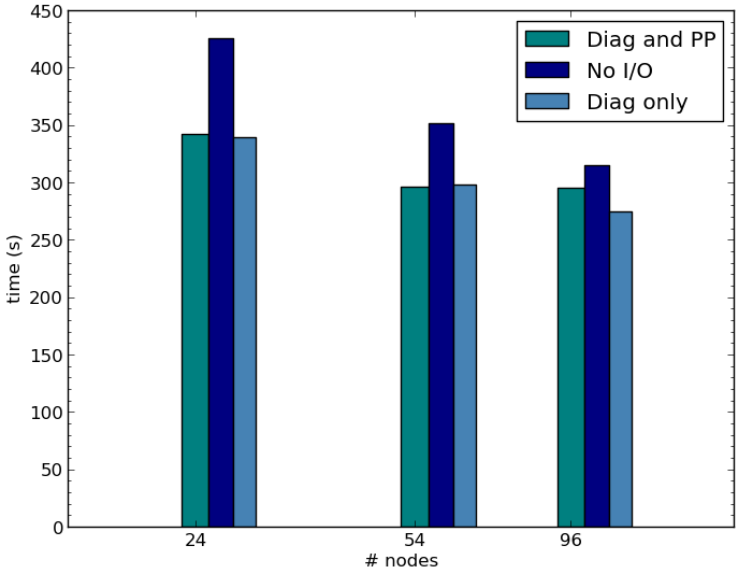
Time-meaning



- Runtime:
  - I/O is asynch. (client wait time close to zero %)
  - No obvious penalty adding time-meaning
  - Runtime variability can be high and obscures scaling signal – needs further analysis
- Memory Usage:
  - There is a modest penalty. About +6-7% for diagnostic I/O over runs with no I/O and a further +1% for adding time-meaning.

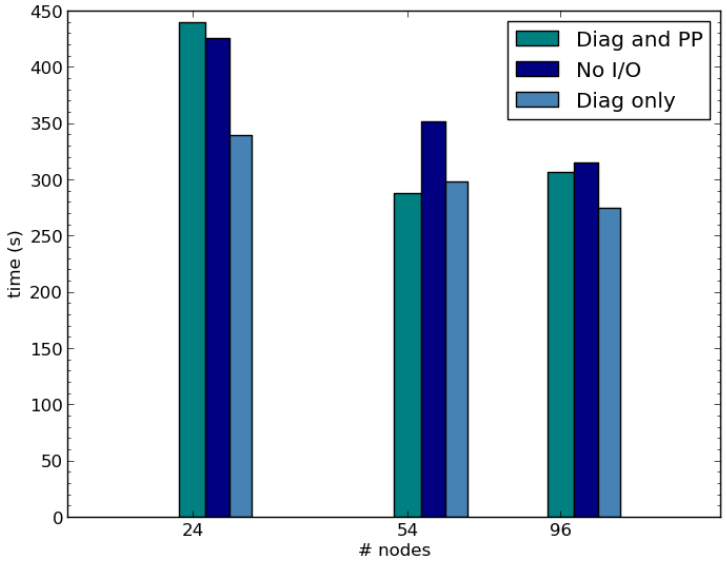
Regridding

- From LFRic native cubed-sphere to lat-lon
- Computing weights on-the-fly
- Runtime:
  - I/O is asynch (client wait time close to zero %)
  - No obvious penalty adding regridding
  - Runtime variability can be high and obscures scaling signal – needs further analysis
- Memory usage similar to time-meaning



Regridding

- Similar expts using pre-computed weights don't show much advantage
- Runtime:
  - I/O is asynch (client wait time close to zero %)
  - No obvious penalty adding regridding
  - Runtime variability can be high and obscures scaling signal – needs further analysis
- Memory usage similar



Next Steps

- Results are encouraging, subject to higher resolution meshes, higher node count jobs of course.
- Probably the benefit of precomputed weights for regridding will be more obvious for a higher res mesh.
- Per-timestep memory monitoring is essential. I had some issues with higher-res and higher node count jobs and I suspect memory issues.
- Need to investigate machine variability for each type of job.

Met Office

Acknowledgements

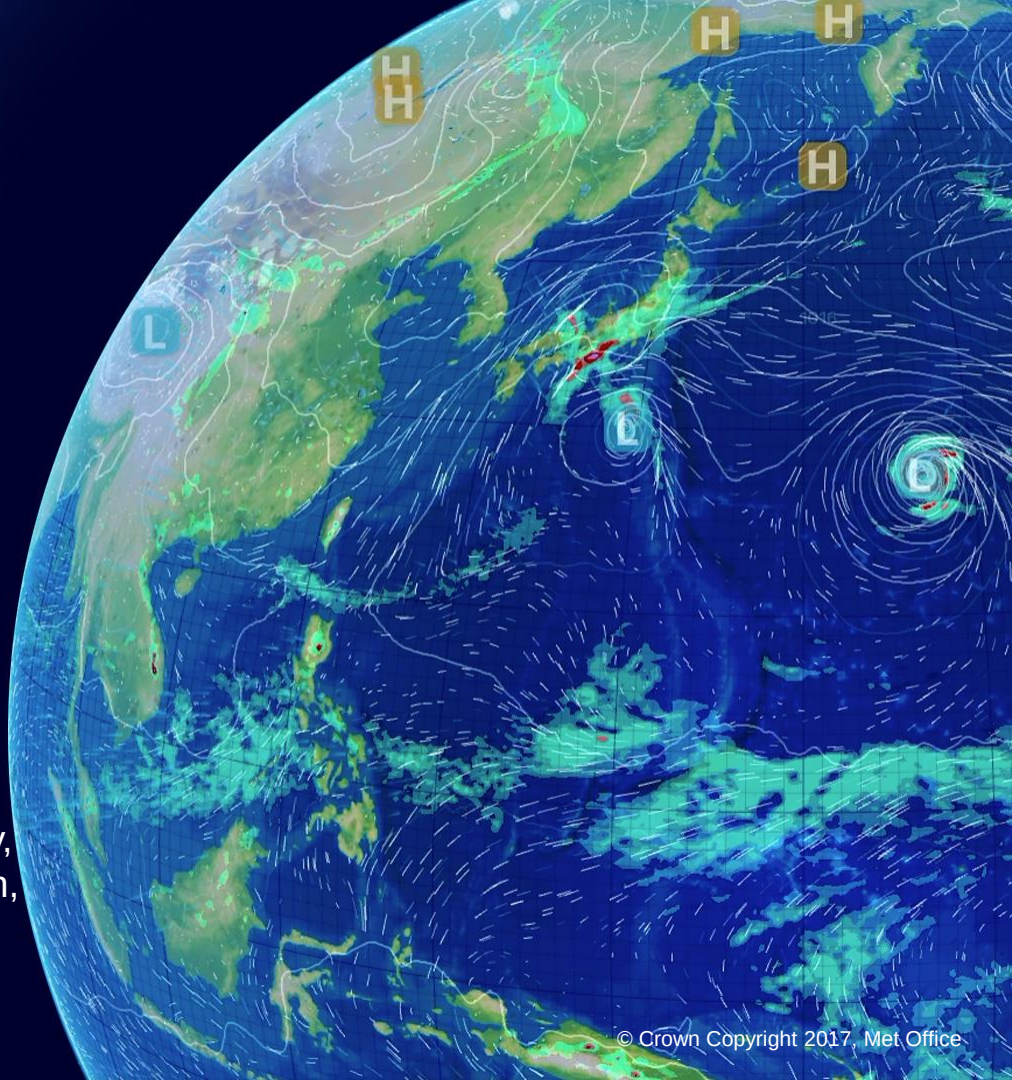
Monash University, **Australia:** Mike Rezny

IPSL (LSCE/CEA), France:
Olga Abramkina, Yann Meurdesoif

NIWA / NESI, NZ:
Wolfgang Hayek, Alex Pletzer

STFC (Hartree Centre), UK:
Rupert Ford, Andy Porter

Met Office UK LFRic team:
Sam Adams, Tommaso Benacchio, Matthew Hambley, Mike Hobson, Iva Kavcic, Chris Maynard, Tom Melvin, Steve Mullerworth, Stephen Pring, Steve Sandbach, Ben Shipway, Ricky Wong



Met Office

Thank You!

Questions?