# Performance Modelling for Climate Models

Computational Modelling Services
NATIONAL CENTRE FOR ATMOSPHERIC SCIENCE
NATURAL ENVIRONMENT RESEARCH COUNCIL

Annette Osprey, Lois Steenman-Clark, Muniyappa Manjunathaiah, Graham Riley

a.osprey@reading.ac.uk

Department of Meteorology
University of Reading

Department of Computer Science
University of Reading

School of Computer Science
University of Manchester

## Building a performance model of the Unified Model

What is a performance model?
- An equation for runtime based on application and machine parameters.

Purpose:
- To quantify and *understand* the behaviour of *current* models.
- To *predict* the behaviour of *future* models on future machines.

Model configuration:
- VN7.3 HadGEM3-A r2.0
- With data from N48 L85, N96 L85, N216 L85

## Motivation

Modelling can be used to explore future application performance in a simulated setting.

This can be useful for:
- Investigating scalability of current models with increased resolution or core counts.
- Predicting performance on future machines which may not even exist yet.
- Exploring algorithm or parallelisation changes, aiding the design process.

The Met Office's Unified Model is a very complex application so it is important to capture it's key characteristics. We follow the approach used to model WRF (Kerbyson and Jones 2007) and POP (Kerbyson *et al.* 2005), deriving an analytical model based on knowledge of the code structure and communication patterns.

## References

T. Davies, M. J. P. Cullen, A. J. Malcolm, M. H. Mawson, A. Staniforth, A. A. White and N. Wood, 2005. "A new dynamical core for the Met Office's global and regional modelling of the atmosphere", *Q. J. R. Meteorol. Soc.* **131**, 1759-1782.

D. J. Kerbyson and P. W. Jones, 2005. "A performance model of the Parallel Ocean Program", *Int. J. High Perf. Comp. Appl.* **19(3)**, 261-276.

D. J. Kerbyson, K. J. Barker and K. Davis, 2007. "Analysis of the Weather Research and Forecasting (WRF) model on large-scale systems", *ParCo 2007*, **15**, 89-98.

P. Selwood and N. Wood et al., 2009. "Strategies for improving the scalability of the UM in response to changing computer architectures", MOSAC meeting paper no 14.10, Met Office.

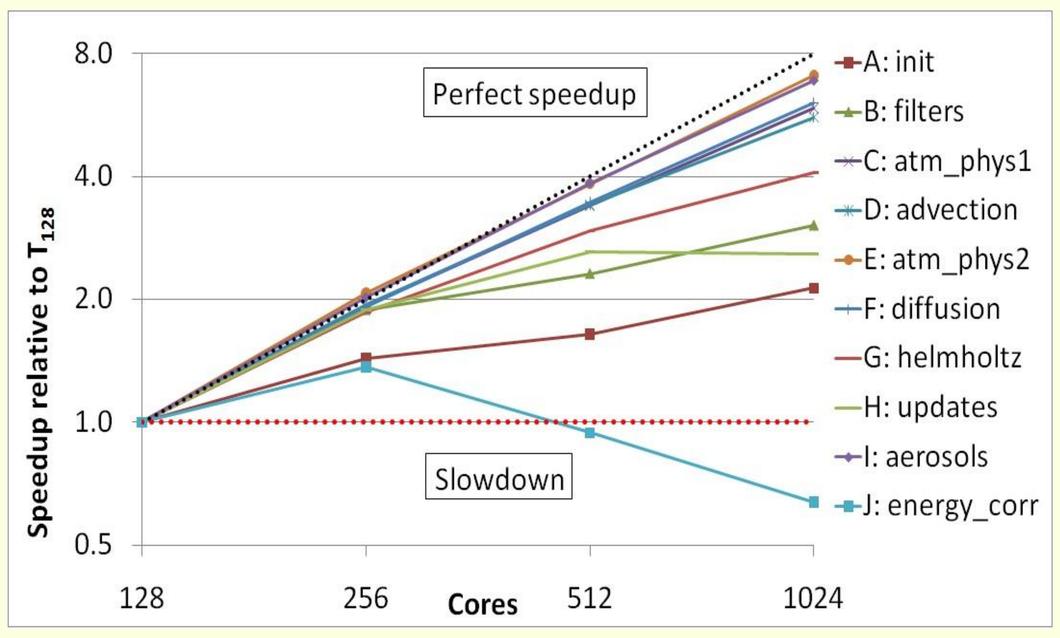
## Breaking down into sub-timestep components

Ignoring the reading and writing of data files, the code structure of the time-stepping loop is:

- Atmospheric science (atm_step)

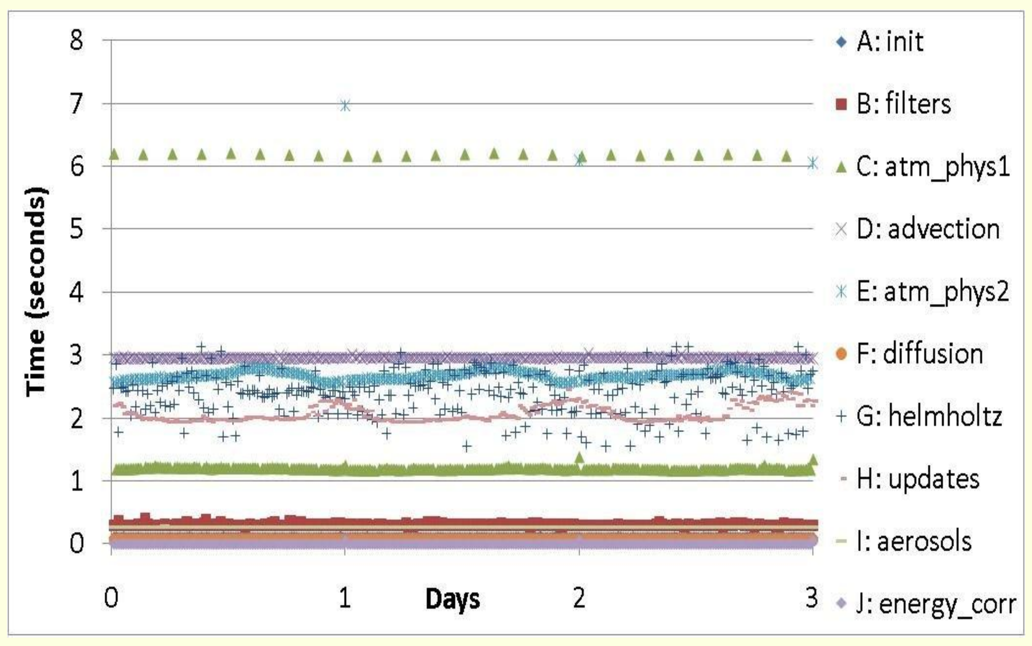
| | | |
|---|---|---|
| A: init | B: filters | C: atm_phys1 |
| D: advection | E: atm_phys2 | F: diffusion |
| G: helmholtz | H: updates | I: aerosols |
| J: energy_corr | | |

Plotting the speedup of each of these sections, we can see that they behave differently.




## Exploring variation over timesteps

Plotting the section times for each timestep of a 3 day run, we can see how the section times change.

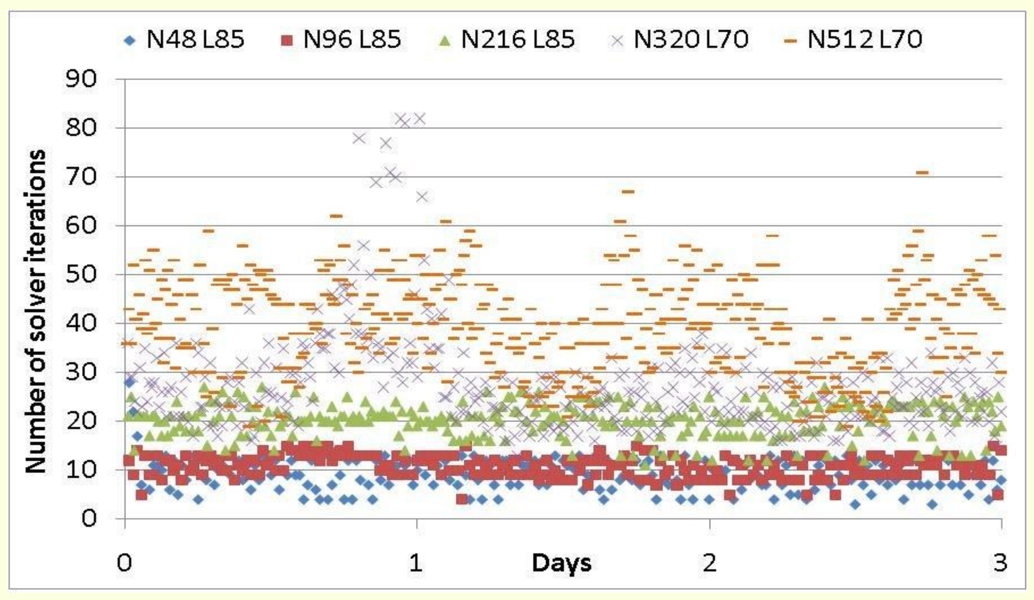



Based on the total runtime we can categorise timesteps into:

- Base
- Radiation

## Exploring variation with resolution

We know that the Helmholtz solver takes more iterations to converge as the problem size increases (Davies *et al.* 2005)




We can use statistics from these runs for the number of iterations per timestep:

| | N48 | N96 | N216 | N320 | N512 |
|---|---|---|---|---|---|
| Mean | 9.3 | 10.8 | 20.2 | 27.7 | 39.8 |
| Min | 3 | 4 | 11 | 13 | 19 |
| Max | 28 | 15 | 27 | 82 | 71 |

## Collecting profiling information

Computations:
- From the UM timers we can derive a mean time per grid point, per timestep for each section.

Communications:
- Identify patterns e.g.
  - halo-exchanges
  - reductions along polar rows
- Where and how often they occur
- Message sizes
- Times

## Putting together a model of the model

The total runtime can be expressed as:

$$T_{total} = N_{ts}\begin{pmatrix} T_A + T_B + T_D + T_E \\ + T_F + T_H + T_I + T_J \end{pmatrix}$$
$$+ N_{ts}(T_{G:base} + N_{itr}T_{G:itr})$$
$$+ N_{base\ ts}T_{C:base} + N_{rad\ ts}T_{C:rad}$$

Each section time ($T_x$) can be split into a communication (MPI) time and computation (non-MPI) time.

$$T_X = T_{comp}(N_x, N_y, N_z)$$
$$+ T_{comm}(P_x, P_y, N_x, N_y, N_z)$$

Where:

- $N_{ts}$ — total number of timesteps
- $N_{base\ ts}$, $N_{rad\ ts}$ — number of base and radiation timesteps
- $N_{itr}$ — number of solver iterations
- $T_{G:base}$, $T_{G:itr}$ — time for non-iterative part of solver code and time per iteration
- $T_{C:base}$, $T_{C:rad}$ — time for physics1 for base and radiation timesteps
- $N_x$, $N_y$, $N_z$ — local data size
- $P_x$, $P_y$ — processor decomposition