



Towards Performance Portability with GungHo

Rupert Ford (1), Matthew Glover (2), David Ham (3), Mike Hobson (2), Chris Maynard (2), Lawrence Mitchell (3), Steve Mullerworth (2), Stephen Pickles (1), Mike Rezny (2), Graham Riley (4), Nigel Wood (2), and Mike Ashworth (1)

(1) STFC, Daresbury Laboratory, U.K., (2) The Met Office, Fitzroy Road, Exeter, U.K., (3) Imperial College, London, U.K., (4) School of Computer Science, University of Manchester, U.K.

The Met Office's numerical weather prediction and climate model code, the Unified Model (UM), is almost 25 years old. Up to the present day the UM has been able to be run efficiently on many of the world's most powerful computers, helping to keep the Met Office at the forefront of climate prediction and weather forecasting.

However, with performance increases from each new generation of computers now being primarily provided by an increase in the amount of parallelism rather than an increase in the clock-speed of the processors themselves, running higher resolutions of the UM now faces the double challenge of code scalability and numerical accuracy.

The UM's atmospheric dynamical core makes use of a finite-difference scheme on a regular latitude-longitude grid. The regular latitude-longitude mesh results in an increasingly disparate grid resolution as the mesh resolution increases due to lines of longitude converging at the poles. For example, a 10km resolution at mid-latitudes would result in a 12m resolution at the poles. The difference in resolution leads to increased communication at the poles and load balance issues which are known to impair scalability; it also leads to issues with numerical accuracy and smaller time-steps due to the difference in scale.

To address this problem the Met Office, NERC and STFC initiated the GungHo project. The primary aim of this project is to deliver a scalable, numerically accurate dynamical core. This dynamical core is scheduled to become operational around the year 2022. The project is currently investigating the use of quasi-uniform meshes, such as triangular, icosahedral and cubed-sphere meshes, using finite element methods.

The associated GungHo software infrastructure is being developed to support multiple meshes and element types thus allowing for future model development. GungHo is also proposing a novel separation of concerns for the software implementation of the dynamical core. This approach distinguishes between three layers: the Algorithm layer, the Kernel layer and the Parallelisation System (PSy) layer. Together this separation is termed PSyKAl.

The Algorithm layer specifies the algorithm that the scientist would like to run (in terms of calls to kernel and infrastructure routines) and logically operates on full fields.

The Kernel layer provides the implementation of the code kernels as

subroutines. These subroutines operate on local fields (a set of elements, a vertical column, or a set of vertical columns, depending on the kernel).

The PSy layer sits in-between the algorithm and kernel layers and its primary role is to provide node-based parallel performance for the target architecture. The PSy layer can be optimised for a particular hardware architecture, such as multi-core, many-core, GPGPUs, or some combination thereof with no change to the algorithm or kernel layer code. This approach therefore offers the potential for portable performance.

Rather than writing the PSy layer manually, the GungHo project is proposing to develop a code generation system which can help a user to optimise the code for a particular architecture (by providing optimisations such as blocking, loop merging, inlining etc), or alternatively, generate the PSy layer automatically.

whilst the PSyKAI approach has been developed for GungHo, it is expected to be more generally applicable. In this talk we will describe the PSyKAI approach, the code generation system and present some early examples of their use.