

## Draft JWCRP Work Plan

F. O'Connor

Version 1.0 – 1 December 2017

Following a discussion with Luke Abraham, Mohit Dalvi, Matthew Glover, Alan Hewitt, Fiona O'Connor, Marc Stringer, and Mark Richardson on the scope and priorities of the JWCRP optimisation work, here is a draft work plan for Chris Dearden setting out a number of tasks for him which also includes some aspects of training on the UM, rose, UKCA and code lodging.

Total effort: 12 months spread over 2 years

1. Become familiar with rose, the UM and UKCA by running the model and doing the online UKCA tutorials
2. Become familiar with the UM code lodging process by lodging a small branch to the UM trunk at the UM June 2018 code freeze e.g. branch to add a print statement which outputs the number of iterations required by the chemical solver to reach convergence
3. Carry out short runs with Dr Hook for profiling purposes – M. Glover to share scripts used by Met Office to help with the interpretation of the Dr Hook output. Gain an understanding of where the model, and in particular, UKCA spends most of its time and how well the model is load balanced.
4. Create a branch to alter the UKCA column call code to split each single vertical column of grid boxes into a flexible number of smaller chunks and alter the corresponding OpenMP directives appropriately.

Assess impact of changes on model cost and investigate the optimum number of grid boxes per chunk (to optimise use of cache).

### Testing strategy:

- Start from a rose suite with no new branch added and with Dr Hook enabled. Run for benchmarking purposes
- Copy suite and add new branch. Then extend this suite to do multiple tasks, where each task runs with a different chunk size through changing a namelist variable. Run all tasks for benchmarking purposes
- Compare model cost before and after change and identify optimum chunk size
- Copy suite with multiple tasks and fix chunk size to optimum size. Then use the multiple tasks to investigate OpenMP performance with different numbers of threads

Lodge the corresponding code changes to the UM trunk with further testing required (e.g. Rose stem, PE bit comparability test, scientific performance etc..)

5. Create a branch to implement a solver compression algorithm such that grid boxes within a chunk that reach convergence aren't re-iterated again alongside those that failed to converge on the first attempt. A branch at an older UM version already exists which could guide this work. Assess impact of compression algorithm on model cost.

Testing strategy:

- Start from a rose suite with no new branch added and with Dr Hook enabled. Run for benchmarking purposes
- Copy suite and add new branch and run for benchmarking purposes
- Compare model cost before and after change

Lodge the corresponding code changes to the UM trunk with further testing required (e.g. Rose stem, PE bit comparability test, scientific performance etc..).

6. If time allows, investigate the possibility of improving model performance through a resource-stealing strategy, thereby reducing the load imbalance of the model – See Cray User Group paper by Glover et al. for details of an approach.