eCSE Final Report for ARCHER2-eCSE03-12

MONC Performance Portability

Grenville Lister[1], Juan Herrera[2], Nick Brown[2], Todd Jones[3], Adrian Hill[4]

[1] National Center for Atmospheric Science, University of Reading
[2] EPCC, UoE
[3] Department of Computer Science, University of Reading
[4] ECMWF

Abstract

MONC is a high-resolution atmospheric modelling code used in the UK by the Met Office and several UK universities. The NERC community wish to leverage the next generation hardware in ARCHER2 for their scientific ambition. This project aimed to update MONC to tune it to the ARCHER2 architecture and take the opportunity to address two long-standing issues, namely, to update the current FFT dependency and relax a restriction in the code that limits the usable halo size.

Keywords: ARCHER2, eCSE, MONC, FFT, halo

Building on ARCHER2

It was recognised that significant exploration would be required to test and develop MONC compiler setting configurations for the available ARCHER2 compile suite. We met with varying degrees of success in our approach to this element of the work.

compiler	**MONC compilation success**	**comment**
cce 11.0.4, 13.0.2	yes	model runs with limited success
gnu 9.3.0	yes	model runs with limited success
gnu 10.2.0, 11.2.0	yes	Unable to resolve rank mismatch errors in `iterativesolver_single_prec_mod` and `iterativesolver_mod` These modules are not required in general. None of the other compilers produce an error in this code.
AOCC 2.2.0.1, 3.2.0	no	Fails compiling `registry.F90` with `Illegal POINTER assignment - type mismatch`
Intel 2021.1 (local install)	yes	Model fails in the IO server

Table 1. Compiler successes on ARCHER2 - there remains some variability in the compilation success rate for the available compiler suite.

FFT replacement

Improve FFTE to handle domains of any size

The MONC model uses FFTE for the Fourier transforms required in the pressure solver. Licensing issues preclude use of the potentially more performant FFTW library (FFTW is licensed under the GPL, and without paying MIT for an unrestricted licence, this is infectious such that those codes which link against it must also be GPL.)

The proposal called for modification of the FFTE library to allow MONC to run on domains of any size. However, after discussion with the FFTE developers, we understood that we would need to engage in a significantly more complex activity than was initially anticipated to develop a generic routine that solves FFTs regardless of the domain size. FFTE only allows domain sizes of a specific nature (divisible by 2, 3, 5) because it takes advantage of algorithms that solve FFTs for this class of domains with a lower number of computational operations.

A search was undertaken for a suitable replacement for FFTE and FFTPACK v4[1] was chosen because the license is free, it was fit for purpose, ie, it solves FFTs regardless of domain size, and its performance is similar to that of FFTE for the 1D decompositions used in MONC. FFTPACK v5 is GPL licensed and therefore was not considered.

Use of FFTPACK allowed us to meet the overarching goal (handle domains of any size), albeit outside the initial idea of "improving" FFTE.

Make FFTE bit reproducible over differing processor decompositions

The MONC test harness implements a series of model configurations to exercise the MONC code base and check results either against known output or against internal criteria. The test harness ran four cases that cover: model dynamics, radiation, micro-physics, and ice in the `rce` (radiative convective equilibrium) case. Each case ran on 36 and 72 mpi tasks and model results are bitwise identical.

Test name	Includes/tests	Bit reproducible
`bubble`	dynamics	yes
`stratus`	SOCRATES radiation scheme	yes
`shallow_convection`	CASIM micro-physcs	yes
`rce`	ice	yes

Table 2. MONC test harness for FFTPACK bit comparison tests - all cases bit compare.

Relax halo size limits

MONC previously only supported halo sizes of 1 or 2. The code has been modified to allow halos of any size. However, we have not yet achieved the complete solution to this work package since halo swapping in the surface boundary condition presents an unresolved problem for halo sizes greater than 2.

The following MONC components have been modified to handle larger halos:

- `decomposition`
- `pressuresource`
- `tvdadvection`
- `lowerbc`

With a halo size greater than 2, changes made `lowerbc` cause the model to fail to converge. Disabling `lowerbc` results in a successful model execution, ie, the model does not fail. Switching off `lowerbc` will basically create a freeslip boundary that is a useful setup for bubble cases or a dry boundary layer case.

From the model core, the following modules are modified:

- `halocommunication`
- `communicationtypes`

A branch[2] with the code modifications is available on the Met Office Science Repository Service.

Performance on ARCHER2

Strong scaling

Strong scaling tests were performed with the GNU case, with respect to a domain of `x_size` = `640`, `y_size = 768` and `moncs_per_io = 15` on fully populated nodes. This corresponds to, for the 128 core case, a decomposition onto 8 IO cores and 120 MONC cores. On each MONC core, there will be 64x64 columns; for a total domain size of 64x64x120 (= 640x768), 491,520 columns in total. In the strong scaling tests, this global domain size is held constant, and the number of cores is increased, keeping the ratio of IO-to-MONC cores constant. The model configuration[3] used is lodged in the Met Office Science Repository Service.

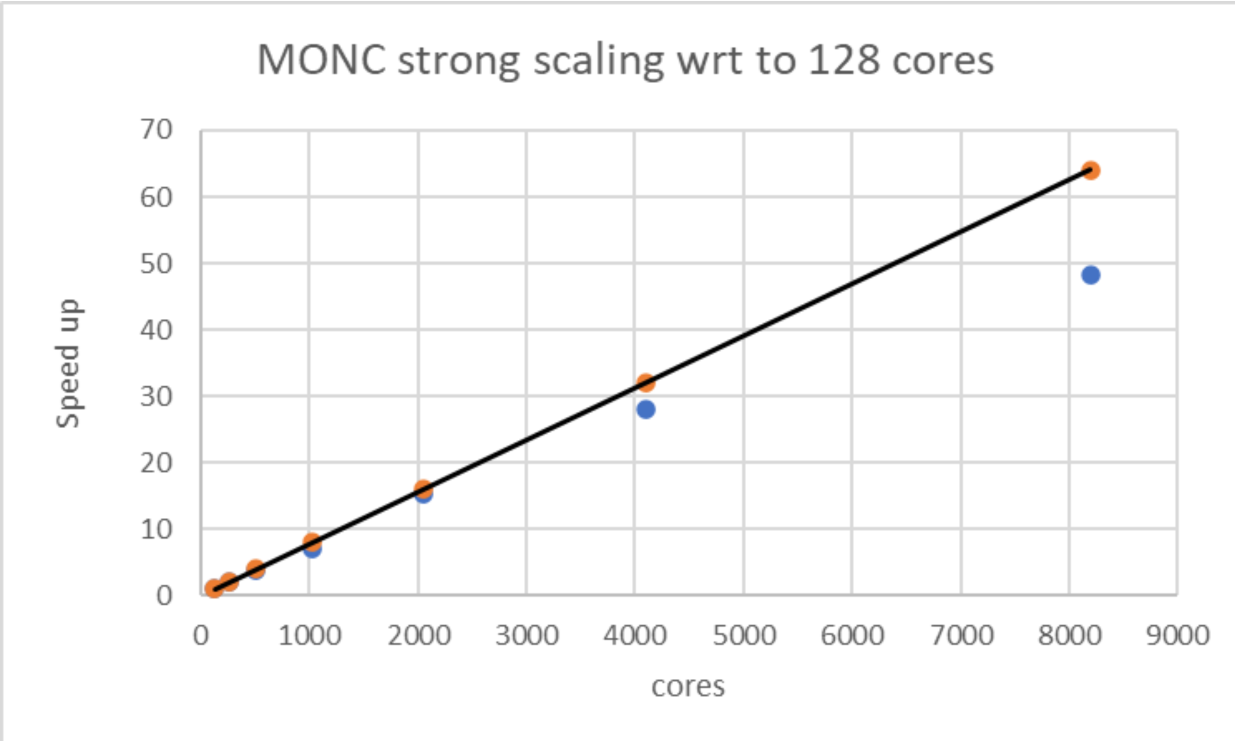

Fig 1. MONC strong scaling. The solid line represents perfect scaling. Blue data points are for the MONC configuration described above.

Weak scaling

The initial configuration is `x_size = 160`, `y_size = 192` and `moncs_per_io = 15` on a fully populated, 128-core node. In this case, there are 16x16 columns on each MONC core, and this computational load is fixed as the number of nodes is increased (i.e., the global domain increases proportionally). The model exhibits good weak scaling for the limited range of cases considered, with small fluctuations in efficiency as node counts increase. We were unable to run the weak scaling test successfully on more than 16 nodes (and encountered an unresolved failure for the gnu case at 8 nodes.) The model configuration[3] used is lodged in the Met Office Science Repository Service.

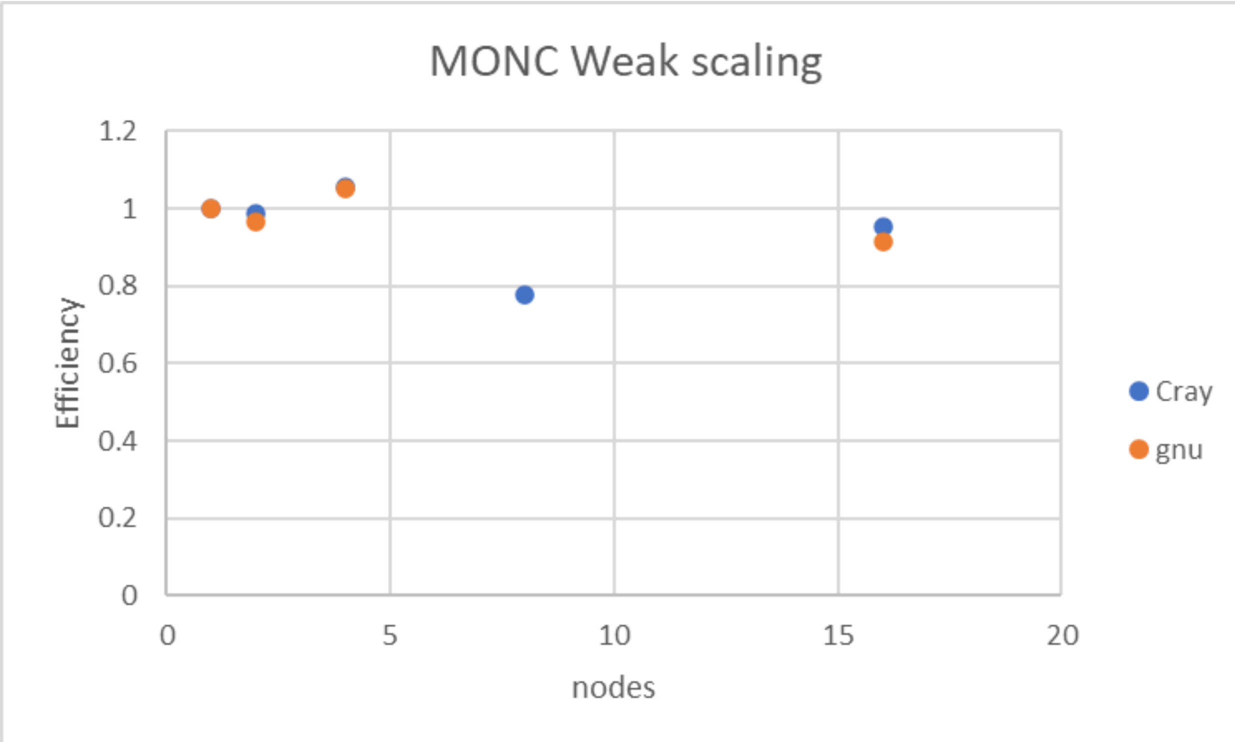

Fig 2. MONC weak scaling. The model exhibits good weak scaling over the limited set of cases that ran successfully.

Conclusion

In this technical report we have summarized the work done under eCSE03-12 in respect of porting MONC to ARCHER2 and addressed some underlying issues that restricted MONC use cases. The code modifications made here are available as branches in the MONC repository. We have met with success in the majority of the developments addressed, namely, removal of dependency on FFTE, licensing contagion, and halo sizing.

Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER2 UK National Supercomputing Service (http://www.archer2.ac.uk). Assistance from NCAS-CMS members David Case and Simon WIlson is acknowledged.

References

1 https://netlib.org/fftpack/
2 https://code.metoffice.gov.uk/svn/monc/main/branches/dev/juanherrera/r9891_halo/
3 https://code.metoffice.gov.uk/svn/monc/main/branches/dev/toddjones/r9891_consolidation_202210