Geosci. Model Dev. Discuss., 7, C447–C449, 2014
www.geosci-model-dev-discuss.net/7/C447/2014/

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "High resolution global climate modelling; the UPSCALE project, a large simulation campaign" *by* M. S. Mizielinski et al.

Anonymous Referee #2

Received and published: 29 April 2014

The paper "High resolution global climate modeling; the UPSCALE project, a large simulation campaign" describes a computational campaign to explore the impact of resolution with the HadGEM3 atmospheric model. This campaign utilized a large 144 million core hour PRACE allocation on HERMIT to generate a series of ensemble runs. The 400 TB UPSCALE dataset generated during this campaign is currently being analyzed as a data archival site in the UK. While this paper is important in that it documents some of the technical and scientific configuration of the campaign it is somewhat disappointing in certain aspects. In particular the computational analysis sections of the paper are disappointing. While I cannot argue for this paper to be decline simply based on the computational analysis sections. However there are a number of opportunities for improvement.

Specific comments:

âĂć It is unclear in section 2.2.3 what they mean by segment size. While this is likely clear to developers of HadGEM3 it is not to an outsider. My guess is that segments are used for cache blocking purposes, however it is not clearly stated. In most codes there are only a limited number of segment size settings that evenly divide the total size of the domain allocated to each MPI task. However, Figure 4 suggests that it can be set to any value. There is a rather ambiguous statement made at the end of page 572 and top of page 573, about segment size and memory management overhead. It is unclear what is meant by "memory management overhead". Another ambiguous statement is made in lines 14-17 of page 573 regarding the reason for performance differences. Please clarify the reason why using different segment sizes impact performance.

âĂć Table 2 suggests that it is possible to use different segment sizes for each section of code. Is this true, most code use a single cache blocking parameter. Why are the optimal setting for segment sizes different for each section of code particularly for the single thread case.

âĂć In section 2.2.4, replace: 25 thousand → 25K or 25,000

âĂć In section 2.2.4 on page 574 a comment is made that production performance of the code does not match the benchmark performance. Several hypothesis are suggested that is could be distribution of nodes, or contention for IO. Well which is it? Can this be determined based on timing statistics from the production runs? If you can't figure this out then say that specifically. The way it is written it leaves the readers expecting an answer.

âĂć In section 2.3 the paper describes the fact that GPS occurred rather frequently during the simulations. How was this addressed during the simulations. Were the simulations restarted from the beginning with a modified time step? Where they rolled back to the last restart and run with a modified time step for a month and then returned to the original time steps. Was grid smoothing applied for a short period of time. It is not clear from the text how GPS were addressed.

âĂć The statement was made at the end of page 574 that the new dynamical core ENDGAME addresses GPS for all resolutions in use. However it is not clear from the text that ENDGAME has been tested at the resolutions used in UPSCALE for future climate scenarios, the configurations that caused the greatest problems.

âĂć It is mentioned on page 577 that there were a large number of job failures. What percentage of jobs failed due to system problems, slow I/O system or numerical instability?

âĂć On page 591, figure 5. A least-squares fit to amdahl's law is provided. However the data to which they fit has different numbers of threads. This prevents the reader form determining the impact of MPI versus OpenMP on the application scalability. I would suggest instead a fit where the number of OpenMP threads is held constant.

Other comments:

The paper mentions on page 577 that bit-reproducibility is important in climate modeling. While I understand this desire and have been aware of it for several decades I am struck by the juxtaposition of this statement and the description of the numerical instabilities that occurred during the simulations. Addressing the numerical instabilities required significant human intervention and some numerical 'tricks'. Without exhaustive documentation of when the numerical tricks were applied to keep the run going means that the simulations are arguably un-reproducible. I understand the justification for the use of such 'tricks' as I have used them myself. Climate modelers really want a statistically reproducible runs which is not the same thing as bit-reproducibility.

Interactive comment on Geosci. Model Dev. Discuss., 7, 563, 2014.