

Interactive comment on “Influence of the compiler on multi-CPU performance of WRFv3” by T. Langkamp

Anonymous Referee #2

Received and published: 3 May 2011

This paper presents a study to assess the impact of different compilers on the performance of the WRF model.

The WRF model has been used by an increasing number of scientists within the last years. Thus the model system has been installed and run on various different platforms using different hardware and software. It is known as well that the model performance can differ between various compilers and compiler options chosen. Different compilers may even lead to differing simulation results and addressing this question would be of interest to a broad community.

Unfortunately I cannot recommend to publish the manuscript in GMD and I will explain in the following why I come to this conclusion:

Overall the results presented in this manuscript are not sufficient for publication in C189

GMD and neither the results nor the methods can be considered as substantially new. The *only* conclusion drawn from this manuscript is the following: The maximum computational efficiency reached for WRF on the *Tornado* cluster was achieved when using the Intel compiler and using 32 cores.

A large emphasis of the paper was laid on discussing scaling issues. Those are however, platform depending questions which are only of interest to people who want to run WRF on Tornado but not to anyone else. It is suggested to find the ideal number of processors for a specific model resolution by simply trying out. This cannot be considered as a scientific method worth publication.

The conclusions drawn in chapter 5 are not at all related to the objective of the paper (compilers and performance). It is no new finding that benchmark tests are helpful.

Even though the paper is categorised as a benchmark paper, having not a single citation included from a peer review journal should not be the standard for publications. The paper could gain scientific importance for the WRF community if it e.g. discussed the impact of different compilers and different optimisation levels on the simulated results.

The manuscript would be as well of more interest if at least an existing benchmarking procedure had been applied consequently and compared to other benchmark tests.

Specific Comments:

– Present the results as well for different optimisation levels (O0-O3)

– It would be nice to add a diagram showing the benchmarking results using different compilers.

– Method of performance measurements (Chapter 3): I would recommend to use the definitions of performance measures as can be found on the WRF benchmark web page. To shorten the description of scaling issues you could present a diagram. Following the suggestions on the WRF benchmark web page you should present as

well what they call *Performance, Floating-point rate, Simulation speed, Scaling*.

– Chapter 4, Results: This chapter has to be rephrased.

– p.560 l 15: Here you describe different configuration setups. All those different setups should be explained once, before you present the results (e.g. in chapter 2.4 after describing the model setup). It might help the reader as well to list the different configurations in a table.

– The benchmark results for the Tornado should be compared with benchmarks on other platforms.

– Can you list major changes to the code between model version 3.1 and 3.2 which might explain the (small) difference in performance?

– You suggest to perform benchmark tests with older WRF versions in the future. I do not really see the need for doing so as the majority of scientist would most likely not use a model version older than WRF v3.

p. 552,553: *However, because it is considered a running system the administrator will not touch it unless absolutely necessary.*

p. 553, l.8: *The installation and benchmarking of another MPI, even of only another version of Open MPI was not a trivial issue.*

I would delete those two sentences. The information is not really of interest to a broader community.

p.553, l. 9-10: *It was partially successful for 10 Open MPI 1.5.1.*

What does 'partially successful' mean?

C191

p. 553, l15-l19: *However, only the latter influences the application's performance significantly. This was verified by turning off all optimizations for the C++ code compilation in the configure.wrf (-O0 instead of -O3 for Intel compilers), which did not change the benchmark results.*

This result is actually not surprising. The c-code (not c++, I think) in WRF is mainly used to create Fortran code which is done once during compilation. All routines which are run during integration are written in Fortran.

Chapter 2.3: I would exclude this section and only refer to the appendix where the reader can find the different compiler settings. Though, it would be of interest to list ALL compiler options (replace the dots with those settings you excluded from the list).

Chapter 2.4: Parts of this section are too specific and only relevant for Tornado users. I recommend to shorten this section and e.g. exclude lines 14-21, p.556

The description of the model setup has to be rewritten. State out clearly which model version with which setup you use and do not write what you are NOT using. List the full namelist.input in the appendix including e.g. the chosen physics and dynamics options. However, I don't think that you need to list the WPS namelist in the appendix.

Chapter 3 has to be completely rephrased.

Again: Do not write what you do NOT do and only write WHAT you do.

Parts of this section contains information only of interest to users of the Tornado (e.g. delete lines 10-12, p.558, shorten or delete lines l22 p.557 - l5 p.558).

Chapter 4:

p.559, l 25,26: *Furthermore the gap between GCC and Intel/PGI decreases with increasing node numbers down to 20% for four nodes or % 32 cores on this system.*

Why ?

C192

p.560 l15-17: You explain the effects of FASTSEE and MVECT, but not MS-MARTALLOC, MPREFETCH, MFPRELAXED

If you want to investigate the impact of different compiler flags on the performance of the code shouldn't you (1) use the same setups for both model versions (2) check the impacts of compiler flags for both other compilers?

Table1, Table2: If you performed the same benchmark tests for both model versions I would wish to have 2 tables with the same content, each one for one model version to be able to directly compare the results.

– I would name chapter 5 only 'Conclusions'.

Language

p.550, l26: Third (instead of 'Thirdly')

p.551, l11: weather prediction model (instead of 'weather prediction system')

p.551, l16: 'WRF ships with' sounds strange to me, I would suggest a formulation like: 'WRF can be run with....'

Interactive comment on Geosci. Model Dev. Discuss., 4, 547, 2011.