Geosci. Model Dev. Discuss., 4, C209–C216, 2011
www.geosci-model-dev-discuss.net/4/C209/2011/

**Geoscientific
Model Development
Discussions**

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

T. Langkamp

thomas.langkamp@uni-hamburg.de

Received and published: 4 May 2011

Dear Reviewer, here my answers to your other comments one by one.

You wrote: "The only conclusion drawn from this manuscript is the following: The maximum computational efïňĄciency reached for WRF on the Tornado cluster was achieved when using the Intel compiler and using 32 cores."

Until you give references, that show that my following additional findings are not new, I can′t agree with this statement:

- On a Cluster with AMD CPUs the compilers of PGI and Intel lead to the same performance over all tested versions. Thus the cheaper of those compilers can be used. The free GCC compiler was significantly slower in the tested configuration, but as J. Kunkel

already mentioned in his comment, an additional test for GCC with proïňĄle guided optimization has to be done.

- There is almost no performance difference between the tested MPI release versions.

- Three different performance fluctuations were detected, discussed in Sects. 2.4 and 3 in more detail. The first fluctuation appeared with a too small computations to communications ratio, suspected to be caused by the switching from InïňĄniband to the slower Gigabit Ethernet under high network loads. It was temporarily circumvented by rising the domains' resolution and limiting the study to 32 cores. The second fluctuation appeared randomly due to the user-induced random load of the network. The third and largest fluctuation appeared periodically at speciïňĄc rush hours before lunch and on Friday afternoons. All three phenomena appeared in a different type of timeïňĆuctuation and were not easy to identify, but are important to know about for accurate measurements. These findings may help others occupied with benchmarking tasks in the future.

You wrote: "A large emphasis of the paper was laid on discussing scaling issues."

Please define "scaling issue" more closely. I used the term only once here: "This optimal gain is possible with the optimal compiler on HPCs, where the network is not the limiting factor, as shown by Shainer et al. (2009) who ran WRF with 192 cores, with the performance scaling up almost linear." Thus I conclude you mean "scaling issue" in terms of "core count", which was discussed in chapters 2.4 and 3 only.

You wrote: "(The scaling issues) 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 ïňĄnd the ideal number of processors for a speciïňĄc model resolution by simply trying out. This cannot be considered as a scientiïňĄc method worth publication."

Just to clarify: In the paper the conclusion of "trying out" was explicitly drawn for the

parameter numtiles and implicitly for the number of core counts, I wrote: “In order to reduce the size of the tiles, it is possible to increase their number via numtiles = x (see Appendix B). However the optimal value x heavily depends on the cache sizes of core and CPU, the number of assigned cores, and the size of the domain. Thus there is no other way than experimentation to find what value gives the best performance.“

I agree that this finding (as I wrote in the paper) depends on the HPC hardware platform which is almost always different. Thus there is no other way than experimenting or “trying out”. Experimenting always was a crucial scientific method often used and it has to be mentioned if it is the method of choice. If you know another method for this case, please let us know. Of course the experiments have to be structured in some way and I can try to point out which structure is the fastest one leading to a satisfactory result.

Your wrote: “The conclusions drawn in chapter 5 are not at all related to the objective of the paper (compilers and performance).”

This is because the results on compilers and WRF-performance were already discussed in chapter 4. Maybe chapter 4 and 5 should be put together.

You wrote: “It is no new finding that benchmark tests are helpful.”

The finding was, that benchmarking of different compilers for WRF is important, as I still believe that there is no peer reviewed literature on this issue.

You wrote: “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.”

I definitely agree, if there are such publications available.

You wrote: “The paper could gain scientiïňĄc importance for the WRF community if it e.g. discussed the impact of different compilers and different optimisation levels on the simulated results.”

I agree partly. I did not know that different compilers with their “safe settings” (which were used only) may lead to different simulation results. Please give some reference on this. Or did you mean the use of “non-safe settings” in terms of non-standard compiler flags that definitely can lead to different simulation results. E. g. Intel writes (in file configure.wrf): “This configuration is aimed at accuracy. To improve performance (at the expence of accuracy) you might consider removing ’-fp-model precise’ flag from FCBASEOPTS. This enables non value-safe optimizations.” Those aggressive optimizations were not considered worth testing, which may be wrong. As this seems to be a new topic I would rather discuss it in a separate paper.

You wrote: “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.”

If “existing benchmarking procedure” means e. g. to measure FLOP/s and other hardware metrics I agree, (as I already answered to J. Kunkel,) because this would generate useful results for compiler programmers. I will look into it for future evaluations.

You wrote: “Present the results as well for different optimisation levels (O0-O3)”

The focus of the paper was to test different compilers and versions without changing the default optimisation levels, because they were considered as the optimum. Why would it be of interest to cut optimal and value-safe optimisations back? This leads to longer simulation times. One positive effect only is the shorter compilation time. But one hour of compilation, which only has to be done once, is nothing compared to the months of time often spent on simulations.

You wrote: “It would be nice to add a diagram showing the benchmarking results using different compilers.”

I will add those.

You wrote: “Method of performance measurements (Chapter 3): I would recommend

to use the deïňĄnitions of performance measures as can be found on the WRF benchmark web page."

If you mean the definition of "Performance" on that page, which is model speed, ignoring I/O and initialization cost, directly measured as the average cost per time step over a representative period of model integration – I wrote that the average cost per time step would not be a sufficient performance measure, as this would mask the optimal reachable compiler performance gain due to the three different encountered performance fluctuations, while the used minimum time for only one time step avoids this shortcoming.

If you mean to use the CONUS benchmark of the WRF benchmark web page instead - I wrote that CONUS lays no emphasis on benchmarking software like compilers, and is to compare different hardware for WRFv3.0. But I wanted to test the environmental software and the more up to date WRF versions 3.1.1 and 3.2.1. Additionally it turned out that the two CONUS benchmarks were not optimal in terms of scaling with core count and needed computing time.

You wrote: "To shorten the description of scaling issues you could present a diagram."

Agreed, it will be added.

You wrote: "Following the suggestions on the WRF benchmark web page you should present as well what they call Performance, Floating-point rate, Simulation speed, Scaling."

For "Performance" see two comments before. For "Floating-point rate" - there was no operation count available to calculate it. "Simulation speed" and "Scaling" will be filled in.

You wrote: "p.560 l 15: Here you describe different conïňĄguration 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 conïňĄgurations in a table."

Agreed to shift it to chapter 2.4. However the table you search for – isn't that Table 2?

You wrote: "The benchmark results for the Tornado should be compared with benchmarks on other platforms."

As I have no access to another HPC (as mentioned in the paper), this has to be done by others, taking the same benchmark setup as I used.

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

All changes are listed here: http://www.mmm.ucar.edu/wrf/users/wrfv3.2/updates-3.2.html. They are numerous. As I am not a programmer I cannot say which one might affect the performance.

You wrote: "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, l15-l19: However, only the latter inïňĆuences the applicationâs performance signiïňĄcantly. This was veriïňĄed by turning off all optimizations for the C++ code compilation in the conïňĄgure.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."

All agreed. I will revise or delete those lines.

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

Partially means that it worked only for one node (8 cores). I will clarify that

You wrote: “Chapter 3: Do not write what you do NOT do and only write WHAT you do.”

I don′t see your point here. Please specify.

You wrote: “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).”

At least lines 10-12 will be of interest on other systems, e.g. where it′s users go to lunch or into the weekend.

I wrote: “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.”

The cause might be network communication overhead.

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

Because they had no effect on performance as written in the paper.

You wrote: “If you want to investigate the impact of different compiler ïňĆags on the performance of the code shouldn’t you (1) use the same setups for both model versions”

I tested different flags only for one version, as there were no performance differences between them, which were significantly bigger than the measuring accuracy.

You wrote: “(2) check the impacts of compiler ïňĆags for both other compilers?”

Different compilers have different flags, thus it is not possible to reproduce the same flag-tests for the other compilers.

You wrote: “Table1, Table2: If you performed the same benchmark tests for both model versions ...”

I did not.

You wrote: “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....”’

All agreed. I will revise those lines.

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