Geosci. Model Dev. Discuss., 8, C4333–C4342, 2016
www.geosci-model-dev-discuss.net/8/C4333/2016/

Geoscientific
Model Development
Discussions

Interactive comment on "Performance and results of the high-resolution biogeochemical model PELAGOS025 within NEMO" *by* I. Epicoco et al.

I. Epicoco et al.

italo.epicoco@unisalento.it

Received and published: 1 April 2016

Dear Referee

Thanks for your comments. Our reply is reported in the following using italic text

> **Specific Comments**
>
> The Introduction is good (sections 1 and 2). I think a bit more detail on the mechanics of the coupling scheme would be useful (accepting that full detail is available in the references). Am I correct in understanding that the BFM and NEMO models exchange data every time-step? Do separate processes execute the BFM and NEMO components or does every PE do

> both components for its subdomain? Please clarify.

We provided in the revised manuscript some additional details on the coupling of NEMO and BFM, by clearly stating that the two models communications occurs at every timestep and each PE execute both components in the associated subdomain (see Sec. 2, line ∼137). However, we still prefer to address the reader to the more comprehensive NEMO-BFM coupling manual (Vichi et al. 2015b, publicly available on the BFM website), which illustrates the data exchange flow and the technical details concerning the realization of the coupling.

> I like the description of the Performance analysis. However, I don't agree with the assertion on lines 252-261 that Figures 2, 3 and 4 show that the MPI communication time is decreasing - those plots show only metrics derived from overall execution time. In fact, it would be nice to see an analysis of the time spent in MPI communication vs the time spent in the rest of the code. Having said that, I agree with the argument that the MPI communication is decreasing - I'd just like to see some evidence. Does the BFM component contribute to the MPI communications or are they only required to support the stencils in the NEMO component?

Figures 2, 3 and 4 were not meant to show the communication time. The considerations about the MPI communication came from some results that were not reported in the paper. Now we added a new chart (Fig. 5) with the communication time for two configurations to give evidence of the decrease in communication time. Even if the computation of biogeochemical processes does not introduce further communications, the BFM contributes to the communications within NEMO since it involves the definition of additional oceanic state variables that are exchanged among processes during the computation of the transport term.

> In section 4 (Code Profiling) it would be nice to have more detail on the % of run-time used by each of the significant routines. Are the entries in Table 3 ordered (e.g. by % of run-time)? A discussion of the differences (if any) of the profiles from the two machines would also be interesting and may highlight opportunities for optimisation. This is a point you touch on later but it would be good to see more detail here.

Table 3 (now Table 4) lists the most relevant routines without any particular order. We added a new Table 3 with the profiling data for both architectures. As reported in the revised manuscript (Sec 4, ln. ∼297), the top 10 routines reported by gprof are the same for both architectures and differences mainly arise in the percentage of running time. On BG/Q we observe also a significant run-time in two system calls for accessing the input files.

> I like the discussion of memory structures in section 5. It is worth emphasising here that NEMO is designed for vector processors and is therefore good for the increasingly wide SIMD instructions that are appearing. How well does BFM SIMD vectorise? Since BFM has zero-dimensional state at each ocean point, presumably it would only need a 2D data structure rather than the 3d one of NEMO (i.e. it has no need of depth information) in order to map onto the topology required by the numerical domain? Would this structure benefit the optimisation of the numerical components of BFM? e.g. remove indirection, promote SIMD vectorisation etc? How much does the load imbalance present in BFM (because of non-ocean points) impact the model as a whole?

In the current coupling structure between NEMO and BFM, the memory of each BFM state variable is defined as a one-dimensional array containing only the ocean points of the entire domain and these are remapped on the 3D ocean grid of NEMO only to compute the transport of biogeochemical tracers. We recognized that the description of the BFM memory layout wasn't sufficiently clear and the related text in Sec. 2 was revised to better explain this aspect. Since BFM arrays are one-dimensional and the solution of biogeochemical processes alone does not require the use of MPI communication, this component of the coupled model is very likely to show increasing performance with increasing number of PEs. However, the load imbalance due to an uneven distribution of ocean points among different PEs is a critical point, as discussed at the end of Section 5.

> I think Section 6 is the weakest in the paper and am unsure of its value since really the key issue is of load balancing performance and thus the ocean points and that is covered in Section 5. Memory consumption is just one symptom of this. However, the data in Figure 11 are intriguing - it seems to indicate that PEs fall into one of three categories and the variation in memory requirements between these is substantial, even for PEs that have very similar numbers of ocean points. I think this deserves further investigation.

Section 6 aims mainly at providing a numerical model to estimate the amount of memory required by each process. The memory model can be used to choose the optimal domain decomposition (i.e. a decomposition such that the memory footprint of the heaviest process is minimum) or it can be used to evenly map each process on computational nodes using the amount of memory per node as criterion. The memory model is not meant to demonstrate the load unbalance of the coupled model. The data reported in Fig.11 (now Fig 13) show that for some process a different amount of memory is allocated even if PEs may have a very similar number of ocean points. This amount of memory does not depend on the number of ocean points nor on the model itself because, with different executions of the same configuration, a given process *sometimes falls in the first category sometimes in an other one. The job_memusage tool, developed by CISL Consulting Services Group at UCAR, has been used to measure the total (peak) memory use of each process, as indicated in the revised version of Sec. 6.*

Technical Corrections

295: describe choice of ordering of entries in the table. Were the key routines the same on the two architectures?

A new table has been added to list the percentage of run time for the most significant routines

305: add the core count information to the caption of Figure 5.

The caption of the figure, now referred as Fig. 6, was updated.

315: suggest replace "are unaffected by scaling" with "do not scale at all".

We modified the text at ln. 357.

316 replace "the results got by" with "the results obtained by"

The text was revised at ln. 358.

327 is that "theoretical" peak performance? What is the theoretical peak performance of a single core?

The theoretical peak performance of a BG/Q core is 12.8Gflops (8 operations per clock cycle at 1.6GHz). One BG/Q node has 16 cores, hence the theoretical peak performance of a BG/Q node is 204.8Gflops. The sustained peak performance, as reported in the top500 list, is 174.7Gflops. Actually we reported a wrong value in the text since the average performance for a whole time-step of the PELAGOS025 model is 0,517Gflops (as reported in table 5) instead of 2.7Gfops. The revised version of Sec. 4.1 includes the above considerations and corrections.

338 replace "routines" with "routine"

We changed the text at ln. 394.

345 I'm not sure that 'random' is right - there must be some deterministic relationship.

Actually 'random' could not be the appropriate term. Even if the computational workload has been efficiently rebalanced from 1344 cores to 2048 cores this is just a 'coincidence' and happens only for the considered decompositions. The whole sentence was revised (Sec 4.2, ln ∼405).

348 "With this architecture, there are more routines with a speedup value far..."

We modified the sentence accordingly to the given suggestion at ln. 409.

Section 5.2: at the risk of self-promotion there is this report http://purl.org/net/epubs/work/63488 which discusses changing the domain decomposition strategy of NEMO in order to load-balance the number of ocean points between processors.

We included the work of (Pickles and Porter, 2012) in the revised text of Sec. 5.2.

> 473: replace "sensible" with "sensitive" (I think "sensible" is strictly correct but "sensitive" is more readily understandable).

We agree with the reviewer and the replacement was done at ln. 534.

> 528-530: this is true if the BFM and NEMO models couple every timestep. A comparison of PEs with large and small memory footprints would be interesting.

As now clearly reported in Sec. 2, BFM and NEMO couple at each time step. Unfortunately we do not have the performance data for comparing the PEs with large and small memory footprint.

> Conclusions, 640. Although a smarter domain decomposition might help at lower process counts, I think Figure 8 shows that by the time you get to 1024 PEs the difference is negligible. Is that right?

We agree with the reviewer comment, since Fig. 8 (now Fig. 9) shows exactly that when the number of PEs is greater than 1024 the difference between both strategies is negligible. We included this comment in the revised section 7 (ln. ∼732).

> **Notes on Tables and Figures.**
>
> As already mentioned, Table 3 does not specify how the routines are ordered. Are they in order of significance (in terms of time taken)?

Table 3 (now Table 4) just lists the routines names and gives a short description of the functionality.

> Table 4: is the "total elapsed time" fully inclusive of start-up costs etc.? If so, some justification that they are negligible is required.

All of the performance values in Table 4 (now Table 5), including the total elapsed time, have been taken without the initialization costs. The "total elapsed time" refers to the execution time of 9 timesteps (from the second timestep to the tenth). The performance measurements do not include the I/O operations (during the first timestep the input files are read and the restart and output writing has been disabled).

> Figures 2, 3 and 4: could be improved by more closely restricting the range of the x-axis to match the range of the data. The data being plotted is not continuous and therefore there need to be points as well as lines on the plots.

The x-axis of these figures was modified into a logarithmic scale to better cope with the wide range of values.

> Figures 5 and 6: I think these plots would be much better as bar charts. The shadows on the points made me think there were two data values at each ordinand at first. As mentioned earlier, please specify the PE counts used in the captions of these figures.

We recognize the reviewer concern and the presentation of data in Figures 5 and 6 is now done using bar chart.

> Figure 9 came out as grey scale for me. If it really is grey scale I suggest colour would really help pick out the differences. Possibly a histogram of number of the number of ocean points on each PE would be useful.

We modified the colorscale of Fig. 9 (now Fig. 10) and a new chart, namely Fig. 11, was added to describe the distribution of ocean points among different PEs.

> Figure 11 - do you have any explanation for the ∼trimodal behaviour seen in this plot? e.g. at very low numbers of Ocean Points some PEs have 0.5 GB allocated while others have 0.7 GB - nearly 50% more. There are yet others with values ∼mid-way between these two extremes.

We measured the allocated memory for different executions of the same configuration; the same process sometimes falls in the first "category", sometimes in the second. Probably some memory allocation happens at system calls level.

> Figure 13 - again data points as well as lines are required on the plot and it would be nice to limit the range of the x-axis a bit more.

We inserted the data points in Figure 13 (now Fig. 15) and limited the range of the x-axis.

Here attached you can find also the pdf with the latexdiff differences between the submitted and revised version of the paper

Thanks again for your comments and contribution

Please also note the supplement to this comment:
http://www.geosci-model-dev-discuss.net/8/C4333/2016/gmdd-8-C4333-2016-supplement.pdf

Interactive comment on Geosci. Model Dev. Discuss., 8, 10585, 2015.