Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-257-AC1, 2020

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on “Parallel I/O in FMS and MOM5” *by* Rui Yang et al.

Rui Yang et al.

rui.yang@anu.edu.au

Received and published: 16 January 2020

Referee’s Comments: The paper describes the implementation and tuning of the parallel I/O in MOM. This is a very good and very timely study, as the parallel I/O continues to be one of the major problems in the current Earth System Science codes. This type of description is usually ether never put together as a text or in a best-case scenario just “collect dust” as a technical report. The authors did a great job of describing in detail their technical development as a paper, and I wish there are more such descriptions in the future. The sharing of this information is very important so that the progress in the field is faster.

Response: We appreciate referee’s comments as above and agree that it is important to share experience and skills in enhancing I/O performance of the climate and earth system models.

Printer-friendly version

Discussion paper

Referee's Comments: I am in general happy with the paper, while having a couple of suggestions that authors free to agree or disagree with. Paper can be published after minor revision. General points To make the paper even more useful it would be nice to discuss several additional details. Short description of how hard it was to implement parallel I/O using each of the libraries (maybe person/month estimate?), what is the user experience with each of the libraries (are they easy to install and support?).

Response: FMS has provided enough functionality to allow us to use the netCDF library directly (please see response to the next comment) while the parallel I/O to netCDF-4 and classic files are achieved through the HDF5 and PnetCDF libraries respectively. All these libraries are widely used in scientific computing; they follow standard installation process (e.g. autootols) and are well supported. It is straightforward to implement I/O operations in the code by invoking the netCDF library APIs after some extra work to set up the I/O domain communicator. The following sentences have been added to the revised manuscript to estimate the time of development work: "Development required approximately one month to implement a working feature, along with an additional month of work to troubleshoot more complex configurations related to land masking and the handling of I/O domains which only cover a subset of the total grid." We also spent plenty of time and effort on the I/O performance tuning, as there are lots of possible bindings among parameters from multiple I/O layers spanning MOM5 I/O domain, netCDF library, MPI-IO and lustre file system. It proves that these efforts are necessary to achieve the optimized parallel I/O performance.

Referee's Comments: Mentioning another parallel I/O solutions, that become popular in Earth Science (e.g. XIOS http://www.ifremer.fr/docmars/html/doc.coupling.xios.html) or even something outside of the ocean modelling world (e.g. https://csmd.ornl.gov/adios) would help the unexperienced reader to be more aware of available software solutions.

Response: XIOS is an attractive I/O framework with the capability to provide highly scalable I/O performance and it has been used in many climate and earth models. We excluded XIOS as the I/O solution in this work because there is a need to maintain the current I/O pattern of FMS in which compute PEs take part in I/O activities rather than setting up the dedicated I/O server with extra PEs as XIOS does. However, the possibility on implementing XIOS in future version is always open. We also ruled out ADIOS as the candidate solution, although it appears to be highly scalable, as it doesn't directly support NetCDF and it requires conversion to NetCDF. It has not yet been used in the weather and climate domain. We have added the following paragraph to Section 2 explaining why FMS was sufficient for our approach. "Because FMS provides access to distributed datasets as well as a mechanism for collecting the data into larger I/O domains for writing to disk, we concluded that FMS already contained much of the functionality provided by existing parallel I/O libraries, and that it would be more efficient to generalize the I/O domain for both writing to files and passing data to a general-purpose IO libraries such as netCDF. In this sense, there is no need to set up the dedicated I/O server with extra PEs as other popular parallel I/O solution like XIOS does."

Referee's Comments: Maybe you can speculate about the applicability of your results to unstructured mesh ocean models, that usually store their results in netCDF as long 1D vectors?

Response: Although we are not very familiar with the implementation details of unstructured models, we believe that there may be a benefit to using the methods detailed here if the data for each mesh element are stored contiguously, and the buffers do not need to be populated from complex data structures associated with the mesh. We have added the following paragraph to the discussion commenting on this issue: "Although this work is applied to a model with a fixed regular grid, these results could be applied to a model with an unstructured mesh. Much of the work required to populate the I/O domains and to define chunked regions is required to produce contiguous streams of data which are passed to the I/O library. If the data is already stored as contiguous 1D arrays, then the task of dividing the data across I/O servers could be trivial. If more complex data structures are used, such as linked lists, then the buffering of data into contiguous arrays could add significant overhead to parallel I/O."

Referee's Comments: Your data-intensive benchmark, although it serves the purpose well, is not very realistic. I think your results will shine even more if you can show how beneficial parallel I/O is in realistic simulations. In Koldunov et al., 2019 we showed that in our case for relatively small setup (about 600 000 surface points) running on 1152 cores the price of the serial I/O in "operational" simulation is only about 5%. For the user that typically has tasks of this size it is not a very large price to pay, and maybe investments in the parallel I/O are not necessary. It would be great if you can run, say, a year of model simulations with typical I/O workload (e.g. in our case its monthly means) on different number of cores with serial and parallel I/O and estimate the amount of time (in %) the I/O takes from the total run time.

Response: We agree with the reviewer that it would be valuable to consider the potential benefits of parallel I/O in more realistic simulations, and have included results from 8-day simulations with 1-day and 4-day I/O frequencies in a new table (Table 6). The serial I/O takes around 6% of total runtime in 720-PE runs which could be regarded as typical I/O workload. The benchmark results indicate that parallel I/O can reduce the I/O time ratio to be less than 1%. More importantly, the serial I/O time ratio of the 1440-PE simulation is about 11%, indicating that the serial I/O may eventually become the parallel performance bottleneck. The parallel write time, on the other hand, scales well with the number of PEs and may prevent I/O from blocking the overall performance scalability.

Referee's Comments: Minor point For figures 3 to 7 please add PE/node as a second x-axis (e.g. on the top). This will make it easier to interpret.

Response: This is a very good suggestion. In this paper the same I/O layout may exist in both 720-PE and 1440-PE simulations. For example, the I/O layout 2*15 could be given by 2 PEs per node*15 nodes in 720-PE or 1 PE per node*30 nodes in 1440-PE simulations respectively. Thus it is impracticable to make the second x-axis on the top as the same I/O layouts may repeat at the first x-axis. Alternatively, we append the PE distribution i.e. [PE per node * nodes] to I/O layout at the x-axis in those revised Figures (Fig. 3∼7) to clarify the connection between I/O layout and PE distribution.

The revised figures 3∼7 and Table 6 are put into the supplement document for reviewing.

Please also note the supplement to this comment:
https://www.geosci-model-dev-discuss.net/gmd-2019-257/gmd-2019-257-AC1-supplement.pdf

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-257, 2019.