Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC1
https://doi.org/10.5194/gmd-2022-77-RC1, 2022

Comment on gmd-2022-77

Anonymous Referee #1

Referee comment on "CIOFC1.0: a common parallel input/output framework based on C-Coupler2.0" by Xinzhu Yu et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2022-77-RC1, 2022

General comments

The paper addresses the highly relevant question of I/O management for Earth System models, by describing the implementation of an I/O server based on the C-Coupler2 library. Taking benefit of the existing remapping and interpolation functions of the coupler, and relying on the XML formalism to precisely define the data to input or output, the authors propose a validated and tested library. This library is essentially made to offer an efficient parallel I/O functionality to the C-Coupler users community.

The authors present a short but comprehensive landscape of the present I/O servers available in the community. The choice of a new development is justified by the opportunity to do it at reduce cost, thanks to a regional cooperation (developers associated to the article) and taking benefit of an existing code (C-Coupler).

As supplementary material, the source code and a user guide is available online. The access to the source code facilitates the understanding of the implementation, even though the article explanations are clear enough to get the overarching strategy . Despite my efforts, it was not possible to compile the test case provided with the source code. The reason is certainly a lack of time that would have been necessary to fully investigate the issues. However, it seems to me that the configuration procedure is rather complex, fitted to the machines architecture of the C-Coupler community and, even though only a small number of external library are required (NetCDF, pnetCDF and MPI), it is difficult to exactly identify the origin of the problem. Sadly, this is a problem that can commonly happen in our community. A simple “makefile” with few compiling option parameters would have been much more convenient …

More generally speaking, such a community tool would be more attractive if its description could prove its usefulness for a larger number of models. This is particularly deceptive in

the case of the CIOFC library because its description shows that very few additional operations are required, starting from a C-Coupler instrumented model, to get the parallel I/O functionality, which makes more attractive, from a developer point of view, the coupler+I/O server suite.

More details would be required to have a better picture of the CIOFC compatibility with a non "C-Coupled" model, e.g. (i) is the tool able to handle masks (sparse matrix) ? or (ii) what are the available interpolations and are they conservative ? The reader who is not familiar with the C-Coupler could be helped if some functions, shared by the CIOFC tool but related to the C-Coupler, could be summarised. Even though it was not possible to reproduce the tests that lead to validate the tool and certify its level of performance, the validation results presented in the document are convincing. One would have like to find there a larger set of performance measurements, not only for competition spirit, but also to be able to evaluate the limits of the chosen technology. However, these limits (synchronicity, single server for the whole ESM components) are mentioned in conclusion, which suggests future fruitful enhancements.

Specific comments

Technical choices are comprehensively described, but one would have like to find more justifications to them. For example:

- the XML format is adopted to select the data that have to be transferred. Even though this a standard choice in the community, I wonder if the XML parser choice and its maintenance could become a problem during the development and would like to know the authors feeling about that.
- same question related to the pnetCDF library maintenance/compatibility with the other netCDF library and compilers. Could the maintenance cost of an additional library be avoided by choosing another I/O server decomposition strategy (one or several global fields per process instead of one 2D horizontal subdomain per process, see for example the 2nd level of server of XIOS v2) ?
- are the output CMOR or CF compliant and if not, why ?

Concerning the launching of the I/O server processes, it is precised that they are considered as a « subset of model processes (p9, l4)». Their number is an XLM file parameter (max_num_pio_proc) but it is certainly also necessary to increase the number of a model process number accordingly ? If yes, on which model should the user do that ? How are these I/O processes identified by the model(s) at initialisation to avoid including them in the pool of its compute processes ?

The procedure which aims to select data from file or model is precisely described, sometimes with too much verbosity, e.g. the one related the output time serie manager (§3.4). It could also be interesting to describe how this information is transmitted to the library (XML parser). How the input/output data of the model is updated could also be unclear to the reader, since no model array to be updated/transferred are provided via the CIOFC API (for writing in output mode, or update in input mode). A check to the example source code shows that it is done via the C-Coupler API, but it could be interesting to mention it in the article.

I also wonder if a CIOFC output or input handler can be set during runtime, or differently said, if the way the model data are modified by input (or the output files) can be changed interactively during simulation ? If yes, can the interpolation be recomputed and if yes, how long does it takes (performance) ?

The authors emphasize the « flexibility to change the source of each boundary field » (p15, l17), but can a variable be switched during runtime from values read in file (input) and read from a model (coupling) ? In another possible configuration, can these two sources contribute simultaneously to define a mixed variable, following a geographical mask (e.g. SST coupled from a model, and lake temperature read in a file?)

For readers interested by comparison (since absolute values are provided in Fig 26-29), it would be good to give more details about the experimental protocol leading to the computing performance measurement at §4.3. For example, the file system kind (and its theoretical performance) could be mentioned. From the model part, the output frequency during the measurement must also be provided.

During measures, the « numerical calculations in real models are neglected» (p18, l3), but are you sure that no interaction between computations and CIOFC mapping could occur ? Other interaction would be interesting to avoid: other users could stress the file system during the test. Was the machine empty during the measurements ? In addition, do you think that three measurements per each test case are enough to neglect the variability induced by the perturbation mentioned above ? If the output frequency is kept constant, how this number was chosen and does its value change the results ? If yes, how ? This could be interesting to understand how much asynchroneous output (or buffering ?) is needed. Could you precise what is the kind of MPI communications in C-Coupler between models and I/O servers (MPI_Send, MPI_Bsend, MPI_Isend ?)

I did not notice any typing errors in the text.