Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-75-SC5, 2020

Geoscientific Model Development Discussions

Interactive comment on "Developing a common, flexible and efficient framework for weakly coupled ensemble data assimilation based on C-Coupler2.0" *by* Chao Sun et al.

Lars Nerger

lars.nerger@awi.de

Received and published: 29 June 2020

Dear Dr. Liu,

Dear Dr. Nerger,

Thanks a lot for taking time to give further clarifications.

We will try to reduce statements or discussions about PDAF and discuss more about the "new aspects" when revising the manuscript, if reviewers agree.

In deed, I hope that my comments will help you, the reviewers and the editor to ensure that no incorrect claims about PDAF will be included in the final

manuscript. They hopefully also help to avoid exaggerations regarding the complexity of tasks like the process setup, the compilation aspect, or adaptions for using all processes in the analysis step.

Here we'd like to reply some of your points.

2: We do believe that these templates can work in most cases, while we note that they should be further adapted for special cases (you may have stated this point in your first round of comment), e.g., the distribution of process IDs among ensemble members is irregular.

It's a design philosophy of PDAF to provide templates that are readily usable for the typical cases, and flexibility to adapt for special cases. This allows us to keep the implementation aspects particularly easy for the typical cases - which are the vast majority by definition of 'typical'.

3: We agree that it would be strange if different ensemble members use different parallel decomposition. For example, would it be a real case that DA is performed for an ensemble of WRF on the same big domain, while only one ensemble member has nested domains? Under such a case, the unique ensemble member with nested domains should use more processor cores so as a different parallel decomposition on the big domain for accelerating integration. As we cannot accurately assert the requirements arising from future model developments, we have to make DAFCC as common as possible.

For ensemble filters one always needs a set of equivalent ensemble members to estimate the ensemble spread or covariance matrix. Thus, the case with only a single ensemble member with nesting is not usable.

4: We do not require the software compilation system of C-Coupler to compile a DA algorithm because dynamic linking is used in DAFCC. When there are a lot of DA algorithms but only using one, only one DA algorithm can be compiled. When integrating a new DA algorithm, the C-Coupler code and mode code can be unchanged. Compilation is one of our consideration while it may not be a critical point when using a framework. We will modify the manuscript accordingly.

Well, your previously suggested statement "... efforts should be made to enable the software compilation system of PDAF to compile the code of the DA methods." is incorrect, because the DA methods are part of PDAF and the claimed effort does simply not exist. When you check the documentation of PDAF, you will see that the compilation is one aspect where we provide recommendations that make this task particularly simple.

5. In short the modification would be as follows: One modifies the parallel setup so that COMM_couple consists of a single process each, which switches off communication in COMM_couple. Then, in the call-back routines (namely in pre-poststep_pdaf.F90) one would add a subroutine to collect the ensemble on the sub- subdomains of the model (thus, on chunks of the state vector) and to distribute the ensemble states again after the analysis update.
Response: Could you please show us the benefit of using PDAF under such a case? Thanks. As the required call-back routines function similarly to the data exchange functionality in DAFCC that is based on C-Coupler, it would be a very challenging work for users to develop such call-back routines. Moreover, further efforts would be required to generate COMM_filter that combines all processes of the whole ensemble of a component model.

The obvious benefit is that PDAF provides the parallelized DA methods. Further, PDAF takes care of the ensemble and the state vectors. Moreover one still has the flexibility of using a different number of processes for the analysis step - the adapted code that I described in my previous comment can also be used for an analysis step using e.g. only the processes of a single model task.
The required call-back routine does actually not need a functionality 'similarly to the data exchange functionality in DAFCC'. For the DA one does not need to develop an actual domain decomposition, but one distributes the state vector that corresponds to a process domain. This approach is much easier than the general domain remapping that one needs in a coupler.
COMM_filter is just the full set of processes. There are no 'further efforts' to generate this (in init_parallel_pdaf.F90 one only needs to change two lines of code and remove one if-clause to obtain the changed COMM_coupled and COMM_filter that I described in my previous comment).

Kind regards,
Lars Nerger

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-75, 2020.