Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-75-SC2, 2020

Geoscientific Model Development Discussions
Open Access
EGU

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.

Li Liu

liuli-cess@tsinghua.edu.cn

Received and published: 30 May 2020

Dear Dr Lars Nerger,

Thanks a lot for your introductions, comments and suggestions.

We are sorry about the misunderstandings regarding PDAF in the manuscript. We will correct them when revising the manuscript based on more discussions with you.

Here, I'd like to briefly introduce the background about DAFCC and this manuscript. We began to design and develop DAFCC in 2017 according to the requirements from operational model development in China. We noted your PDAF work after someone

Printer-friendly version

Discussion paper

introduced it to us in 2018. Then we know that PDAF is a pioneer of our work especially after your latest GMDD manuscript was online. To make DAFCC known by potential users and to share some "new" aspects in developing a DA framework, we submitted this manuscript to GMD for possible publication. We had to state the differences between PDAF and DAFCC because it will be easily asked what are new advancements from the state of the art in the review process.

The statements about PDAF in the manuscript are from our understandings based on the documentations, papers and existing code versions (including the latest public version V1.15.1) of PDAF. It is a wrong statement that PDAF requires each ensemble member use the processes with successive IDs in the MPI_COMM_WORLD, because the communicators COMM_filter, COMM_model and COMM_couple are generated by the corresponding user code. We will correct this statement when revising the manuscript. There would be differences between PDAF and DAFCC regarding the generation of communicators, where PDAF relies on users' efforts while DAFCC automatically generates communicators implicitly. According to our experiences from the cooperation with Chinese model teams, we just feel that, it is not easy to develop the codes for generating the communicator of the ensemble of a component model in the ensemble run of a coupled model, especially for scientists.

We inferred that PDAF requires all model ensemble members use the same number of processes and the same parallel decomposition, and only makes the processor cores of the first ensemble member available to the DA algorithm, based on the PDAF codes (e.g., version V1.15.1). For example, in the "SUBROUTINE PDAF_get_state" in the PDAF code file "PDAF-D_get_state.F90", the root process in COMM_couple corresponding to one ensemble member gets state variables from the remaining processes in COMM_couple corresponding to other ensemble members. Based on the examples available in the code package, we know that the global communicator is generally split into a set of COMM_couple each of which corresponds to the ith process of all ensemble members. We really do not know how to organize COMM_couple and whether

"PDAF_get_state" still works, under the case that model ensemble members use different numbers of processes or different parallel decompositions, or a DA algorithm uses all processes of the ensemble. Could you please show some examples about that case? Thanks.

We are sorry of that "... efforts should be made to enable the software compilation system of the model to compile the code of the DA methods." is incorrect. How about "... efforts should be made to enable the software compilation system of PDAF to compile the code of the DA methods."

Regarding your comments "However, implementing them is a pretty trivial task. Actually, this operation is always model-specific and an analogous operation even happens when using a coupler like C-Coupler. However, in this case it's in the coupler API instead of the assimilation API.". In our opinion, it will be not a pretty trivial task but a heavy task even like developing a new coupler when the DA algorithm uses a different parallel decomposition. C-Coupler has formalized model-specific operations for data transfer among different component models or different parallel decompositions with standard APIs, like other couplers. So DAFCC that is based on C-Coupler2 does not require users to conduct such tasks.

Regarding your comments "Thus by letting more processes compute the DA algorithm, the overall speedup will be limited. In contrast, using more processes for the analysis step requires remapping of the domain decompositions. This requires more MPI communication calls, which will take more time than just collecting ensemble information on the first ensemble task. Thus, while one gains speed in the analysis one looses time in the remapping. What is faster will be case-dependent.". We agree that what is faster will be case-dependent. That's why we try to offer a maximum number of processes to DA algorithms while a DA algorithm can only use a part of processes it can effectively use in real cases. Could you please show us a detail example how PDAF enables a DA algorithm to use all processes in the ensemble of a component model (the DA algorithm will generally use a very different parallel decomposition from the

model ensemble), without modifying the PDAF codes (e.g., version V1.15.1) and with trivial efforts in developing call back functions.

Wish more discussions with you. Then we can list out the new statements regarding PDAF. Many thanks again.

Best regards,

Li Liu

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