Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-27-RC1, 2018

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting" *by* Li Liu et al.

Anonymous Referee #1

Received and published: 9 March 2018

Title: C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting Author(s): Li Liu et al. MS No.: gmd-2018-27

General Comments:

This paper presents an overview of the updated C-Coupler, C-Coupler2. Overall, it is well written and provides both the motivation and implementation details for the updated version of the model. This paper is primarily am overview paper and it does not contain much application or performance results, but I feel this is reasonable at this time. I would encourage the authors to publish additional results in the future detailing the performance cost of higher resolution and higher core counts tests and sharing the performance of 3D weight generation and coupling.

Specific Comments:

page 23, lines 25-29. The ability to run with lags properly is critical. Lags almost always create additional requirements on restart files and the ability to restart a model exactly (bit-for-bit) with lagged coupling fields should be a requirement if lags are going to be supported. It sounds like this is not currently supported in C-Coupler2? Maybe rather than saying "We therefore propose", it would be clearer to say something like "Lags are not fully supported in the current version ,but in the future, the C-Coupler2 will . . ."

page 26, Section 4.5. Is the only reason MPI_Put and MPI_Get is used is to avoid possible exhaustion of MPI buffer space? That should be very rare in practice. Are there other reasons? Performance, ease of implementation, etc? Based on the description on Section 4.5, the MPI_Put/Get implementation sounds slower and more complicated than well managed MPI_ISend/IRecv implementations with MPI_Wait implemented appropriately. Are the authors happy with this implementation? Section 5.3 answers this question in part, but it might be nice to add a few more words in either section 4.5 or 5.3. I think one-side communication potentially helps with both MPI buffer usage and ability to have greater flexibility in coupling lags, but does not improve performance? How about implementation complexity?

page 28, line 8. This is a nice feature. One has to be concerned about memory usage but this provides a nice way to allow extra flexibility in lags compared to other implementations.

page 22, paragraph beginning at line 6. I believe the comparison between Oasis3-MCT_3.0, CESM, and C-Coupler2 is not particularly clear. The authors compare how components interact in different systems, but the definition of the component is not the same in each system. In CESM and the C-Coupler2, the component is defined by separation of scientific models. In Oasis, the component is defined by the separation of MPI tasks. In addition, CESM is more than a coupling layer, it also includes a top level driver that supports the ability to call multiple components from the same MPI tasks in a single executable but only to couple via the driver layer. Oasis3-MCT_3.0 does not have a driver layer and is driven by calls from inside the models. In practice, users could implement a top level driver using Oasis3-MCT_3.0, so Oasis3-MCT_3.0 can behave just like CESM plus it can behave in other ways. I am still a little unclear about whether the C-Coupler2 consists of a driver. If so, is it just a single executable system or does is support multiple executables? I believe none of the coupled systems discussed in this paragraph support multiple MPI tasks running on a single processor, and otherwise they are very similar in capabilities. The main difference is that CESM does not support coupling within a component compared to the other two. I think this paragraph should be clarified. It's difficult to read and the similarities and differences should be more clearly qualified.

page 22, paragraph beginning at line 22. It seems C-Coupler2 is using a file to coordinate MPI tasks between components. While this may be simpler than synchronizing with MPI, there is still the equivalent of a global barrier in the interaction. A component cannot know the tasks of other components until other components have written to the file. How does the C-Coupler2 ensure that other components have written to the file before the information is needed? What is "file" synchronization chosen over MPI?

page 34, line 30. The issues with 3D conservative coupling are the same as 2D. Even with areas, model areas and conservation method areas can differ and this needs to be taken into account with 2D conservative mapping. I do no believe there are any fundamental hurdles to extend 2D conservative coupling to 3D and there may be tools that already accomplish that.

Is the C-Coupler a hub coupler a component, is it just a layer in the system, is it the driver? I think C-Coupler1 was a hub and C-Coupler2 is a coupling layer, is that correct? It might be good to discuss this in the introduction and in regard to Figure 1.

With self-coupling or self-nesting on the same pes with the same executable and multiple grids, how does the C-Coupler2 address the issue of multi-instance data privacy

Printer-friendly version

Discussion paper

within the executable? It may not be enough just to instantiate a new domain or a new state. The underlying model has to meet specific and complex requirements to support that feature with regard to fully separating the memory of the two instances and most models do not. Does the C-Coupler2 actually support this and does it introduce any requirements on components to support that capability? For example, running multiple instances on concurrent pes does not create the same problems. Also, using the C-Coupler2 to couple internal data within a model that supports nesting is not difficult. It's not clear whether the C-Coupler2 supports self nesting on overlapping pes between a component model and another instance of the same component model. Section 4.6 suggests it can. How can that be? Maybe that could be clarified. This comes up in Section 3.6 and Section 4.6.

Does the C-Coupler2 support unstructured grids in 2D or 3D such as cubed sphere, non quadrilaterals, and other complex geometries? Does the on-line remapping support weight generation for those grids? Please indicate in the text.

The results show reasonable performance at moderately high resolution and pe counts. I think these results are adequate at this point, but it would be nice if there were an opportunity to test and publish results at higher resolution and higher task counts in the future, and I agree with the final statement on page 35, line 6.

I think section 4.1.1.1 to 4.1.1.8 could be removed and the user guide could be referenced instead. I think the API details are not needed in this paper. 4.1.1 could just a paragraph that provides a few sentences about the API and points to the user guide plus 4.1.1.9. That would be my recommendation, but will allow the authors to respond to this point.

Technical Comments:

Use of word “generations”. Maybe it can be defined in first use as it is not clear or maybe another word is better, like “coupling interactions”, “coupling procedures”, or “coupling methods”. I realize “coupling generations” is the output of the “coupling gen-

Printer-friendly version

Discussion paper

erator" but it's not the clearest language.

page 4, line 9, please define CoR1 better at the first instance and as needed in other locations in the paper and provide a reference if it exists.

page 5, line 14, chemistry can also be a separate package/component and 3D coupling in that case is important. I don't think you should say "is always included as an internal package".

page 5, line 28, please clarify that time varying is only supported in the vertical dimension, not the full 3D grid.

page 6, line 25, most couplers non block on sends and block on recvs to reduce deadlocking. Deadlocking is always an issue even for fully non-blocking communication. At some point, you have to block and check the data has been received before it's used.

page 10, line 13, locates is not a good word, try "a gridcell"

page 12, line 10, please clarify "model whose model coupling is fully served by other couplers but not C-Coupler2 is unnecessarily registered to C-Coupler2". Does this mean is should be registered, should not be registered, can be registered, or what? This sentence is unclear.

page 13, line 3, I find use of "timer" in this context to be confusing. I think you mean time, coupling period, alarm, coupling frequency or something similar. I understand timer is the word you have chosen to use in the interface, but it would be good to explain what "periodic timer" is in the context of the C-Coupler2. I think it defines the coupling period/frequency?

page 15, line 4, what is buf_mark?

page 23, line 23, please rewrite the first sentence in this paragraph. It is unclear.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-27, 2018.

Printer-friendly version

Discussion paper