Geosci. Model Dev. Discuss., 7, C1009–C1014, 2014
www.geosci-model-dev-discuss.net/7/C1009/2014/

Geoscientific Model Development Discussions
Open Access

Interactive comment on "C-Coupler1: a Chinese community coupler for Earth System Modelling" *by* L. Liu et al.

L. Liu et al.

liuli-cess@tsinghua.edu.cn

Received and published: 29 June 2014

We thank Reviewer #1 for the comments and suggestions. We will modify the manuscript according to them in the revision process. In the following, we will reply them one by one.

1. There is little discussion of technical details such as how coupling fields are defined in the system, how they are reconciled between models, how models are advanced in time in an orderly fashion, or how sequencing/concurrency/lags are established between models. Some discussion needs to be added of these aspects.

Response:

1) How coupling fields are defined in the system?

Full Screen / Esc
Printer-friendly Version
Interactive Discussion
Discussion Paper

A coupling field, such as SST, can have different instances in the component models due to different parallel decompositions and different grids. In the C-Coupler, there is a keyword for each field instance, which consists of the name of the field, the name of the corresponding model, the name of the corresponding parallel decomposition and the name of the corresponding grid. To define a field instance, the corresponding four names must have be defined or registered to the runtime software system of the C-Coupler. All legal field names are listed in the configuration files, with other attributes of the fields, such as the long name (also known as description) and the unit. A parallel decomposition is defined by a component model through calling the API c_coupler_register_decomposition. A grid is defined in the corresponding CoR script. For the scalar field which is not on a grid, name of the corresponding parallel decomposition and grid is labeled "NULL". When a component model registers an external field to the C-Coupler, these four names are specified when calling the API c_coupler_register_model_data.

2) how they are reconciled between models

All instances of a coupling field share the same field name. The field values in one instance can be transformed into another instance through data transferring between two component models and data interpolation intra a component model. All component models in a coupled model share the names of the coupling fields.

3) how models are advanced in time in an orderly fashion, or how sequencing/concurrency/lags are established between models

The C-Coupler1 implements the timer manager (pg 3908: Line 25) as well as related APIs to make components in a coupled model advance in time cooperatively. Each component model calls the API c_coupler_advance_timer to advance the simulation time. An algorithm or an operation is executed only when the corresponding timer is on. "The count of delay" in the timer specifies the lag for model coupling. Sequential and concurrent runs between component models can be achieved through cooperatively

GMDD
7, C1009–C1014, 2014

setting the delay of the timers.

2. Can the C-Coupler infrastructure handle unstructured grids?

Response: The C-Coupler can use the remapping weights generated by the software CoR. Compared to the software SCRIP, the CoR can handle more kinds of grids. For example, when using bilinear remapping algorithm in the SCRIP, the source grid cannot be a cubic spherical grid, while the CoR can handle this case. Therefore, the C-Coupler can handle some unstructured grids. The 2-D remapping weights generated by the CoR can also be used by other couplers because the weight values can be formatted in the "SCRIP format".

3. It's suggested in the description and implementation that CPL6 served as a starting point for much of the work ... what is required of new components to allow them to couple with the C-Coupler?

Response: Generally, it takes several steps to use the C-Coupler to couple a component model that has already been coupled with the CPL6: 1) generate remapping weights if necessary; 2) write a CoR script to register the grids of the component model and read in the remapping weights; 3) initialize the C-Coupler runtime software system and get the MPI communicator through calling API c_coupler_initialize; 4) finalize the runtime software system through calling API c_coupler_finalize to; 5) register each parallel decomposition to the C-Coupler through calling API c_coupler_register_decomposition; 6) register each field instance through calling API c_coupler_register_model_data; 7) call API c_coupler_execute_procedure to send and receive coupling fields; 8) write configuration files (section 4.2.1) of the component model, to integrate the component model into the C-Coupler platform. We note that similar steps are required when coupling a new component model with the CPL6. Therefore, it is not difficult to use the C-Coupler to couple a component model with CPL6 APIs. To construct a new coupled model with new component models, there are further steps required: register external algorithms if necessary and write the

GMDD

7, C1009–C1014, 2014

configuration files for the coupling procedures (e.g., how to transfer fields, how to interpolate data and how to calculate flux). In the C-Coupler1, the configuration files for the coupling procedures are written manually by scientists. In the future C-Coupler2, they will be generated automatically by the coupling generator.

4. The implementation seems extremely flexible but also quite complex. Can the authors provide some addition insight regarding what's working well and what is more difficult than expected. How difficult is the C-Coupler to use and how steep is the learning curve? How robust is the system?

Response: The most remarkable difficulty for us when developing the C-Coupler is the implementation for 3-D interpolation. The challenges include: 1) how to manage the 3-D remapping weights. For the "2-D+1-D" implementation of the 3-D interpolation, the 2-D and 1-D remapping weights should be managed separately. Generally, there is only one matrix for the 2-D remapping weights. However, considering the 3-D ocean mask when z grid is used for the vertical direction of the ocean grid, there possibly are multiple 2-D matrixes, each of which is corresponding to the horizontal 2-D grid on a vertical level. Similarly, there could be multiple matrixes for the 1-D remapping on the vertical direction. 2) How to store the 3-D remapping weights in file. Similar to other couplers, the C-Coupler1 supports to use the offline remapping weights read from file, in order for reusing the remapping weights and saving the time for generating the remapping weights. It is very difficult to store the 3-D remapping weights in Netcdf formatted file. Therefore, we made the CoR to store the 3-D remapping weights in binary file.

To construct a coupled model with the C-Coupler1, users have to study how to write the configuration files for the runtime algorithms and runtime procedures. The configuration files are in specific ASCII formats. Users must spend some time to study these ASCII formats. This may be difficult to C-Coupler users. In the future version C-Coupler2, these configuration files will be generated automatically by the coupling generator, which will facilitate the use of the C-Coupler.

GMDD
7, C1009–C1014, 2014

We think that the C-Coupler is robust enough for use. There are a lot of diagnostic code segments in code of the C-Coupler, e.g., one diagnostic code segment per 35 source code lines.

5. How expensive is the remapping weights generation in the C-Coupler, is it parallel, and does the performance scale?

Response: The C-Coupler can use offline remapping weights generated by the software SCRIP and CoR. Considering that the vertical grids (such as SIGMA-P grid) of component models may be changed during the model execution, the 1D remapping weights for the "2-D+1-D" interpolation can be generated online in parallel. This online weight generation can scale well because the vertical grid is not decomposed for parallelization and each process can generate the 1D remapping weights independently.

6. It would be interesting to discuss how the 3-D interpolation is setup. I expect it's just a large linear weights matrix. Please clarify that point. How is the vertical spline interpolation handled? How will further non-linear or equation solving be handled? Will those methods scale well?

Response: As introduced in the answer for question 4, the 2-D and 1-D remapping weights for the 3-D interpolation are managed separately, and there could be a number of matrixes of 2-D remapping weights and a number of matrixes for 1-D remapping weights. It is possible to merge all matrixes for 2-D and 1-D remapping weights into one matrix of 3-D remapping weights. We do not select this implementation because it may increase the calculation for 3-D interpolation. Moreover, this implementation cannot handle the vertical spline interpolation which requires equation solving. For each spline interpolation on the vertical direction, the coefficient matrix of the equations is pre-calculated by the CoR when generating the offline remapping weights. During the model execution, the C-Coupler will call the CoR functions to solve the equations. The vertical interpolation scales well because the vertical grid is not decomposed for parallelization and each process can handle the vertical interpolation independently.

GMDD

7, C1009–C1014, 2014

7. Is it possible to provide some more content about scaling ... Have tests been carried out on large (ie. high resolution) grids? Please provide additional technical details in the performance results. In 5.5.1 and 5.5.2, please clarify what is being coupled/remapped ... It's typical for the scaling to level off or even turn over for communication dominated kernels, please show that.

Response: During the development of the C-Coupler1, we parallelized the runtime software system with MPI but rarely optimized the parallel performance. However, we will show more parallel performance evaluation according to Reviewer #1's suggestions when revising the manuscript. We think that, there will be the case of the scaling to level off even turn over due to the rapid increase of the communication overhead.

8. Technical Corrections

Response: We thank Reviewer #1 for giving so many suggestions. We will correct the syntax errors and invite an English speaker to help revise the manuscript.

Interactive comment on Geosci. Model Dev. Discuss., 7, 3889, 2014.