Parallel Computational Fluid Dynamics: Theory and Applications

A. Deane et al. (*Editors*)

A Generic Coupler for Earth System Models

Shujia Zhou,[a] Joseph Spahr[b]

[a] [*]*Software Integration and Visualization Office, NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA*

[b]*Dept. of Atmospheric and Oceanic Sciences, UCLA, Los Angeles, CA 90095, USA*

Keywords: Coupler; Earth System Modeling Framework; distributed data broker

Abstract Although a standard way of exchanging data between models deployed by the Earth System Modeling Framework (ESMF) facilitates coupling of Earth system models across organizations, a coupler has to be custom-built for each application owing to lack of a data registration service. In this paper, we will discuss the generic coupling features used in Earth system models and present a generic coupler for certain cases.

1. Introduction

One of the distinct features of Earth system models (e.g., atmosphere, ocean, land, sea-ice) is that the models need to exchange data, typically along the model interface and between different grids, to obtain an accurate prediction. In addition, these Earth system models typically run on parallel computers, at least in the production mode. Therefore,

[*] Employee of Northrop Grumman Corporation.

various parallel software tools have been developed to support coupling operations such as the UCLA Distributed Data Broker (DDB) [1], the NASA Goddard Earth Modeling System (GEMS) [2], the GFDL Flexible Model System (FMS) [3], and the NCAR Flux Coupler [4]. Although those tools have been used in production, they are typically used only within their own organizations. It still remains challenging to couple models across organizations.

The Earth System Modeling Framework (ESMF) is funded by NASA's Earth-Sun System Technology Office/Computational Technologies Project to facilitate coupling Earth system models across organizations [5, 6]. So far, several cross-organization, sequential model couplings have been successfully enabled by ESMF [7]. That is partly because ESMF defines a way of coupling models based on the consensus of the Earth system model community and provides utilities such as parallel regridding. However, a coupler has to be custom-built for each coupled system, which hinders the usage of ESMF.

In addition, we find that there are a lot of similarities among the couplers for ESMF-enabled coupled Earth system models. In particular, the couplers for the combinations of the UCLA atmosphere model [8] with the LANL POP ocean model [9] and the MITgcm ocean model [10] are the same, except for the name and the number of exchanged variables. Couplers for the NASA-NCAR fvCAM atmosphere-NCEP SSI analysis [11, 12] and the GFDL FMS B-grid atmosphere-MITgcm ocean combinations are nearly identical as well [3, 10].

In this paper we will discuss the generic coupling features used in Earth system models and introduce a data registration service to enable a generic coupler for the cases in which regridding will only be considered (e.g., not unit conversion).

2. Model coupling issues

There are four major issues associated with the coupling of Earth system models:

1. The function (subroutine) names in model components are different.
2. The names of import variables of a “consumer” component are different from the names of export variables of a “producer” component.
3. The number of import variables of a “consumer” component is different from the number of export variables of a “producer” component.
4. Maintaining a consistent inter-component list of the data elements to be transferred between components.

To cope with the first issue, ESMF standardizes three basic operations in Earth system models with ESMF_GridCompInitialize(), ESMF_GridCompRun(), and ESMF_GridCompFinalize(). Each interface also has a stage option. In addition, ESMF provides a “producer” component with a function registration service, ESMF_GridCompSetEntryPoint(), to link its subroutines to those three standard

interfaces, and provides the "consumer" component with an accessing service, ESMF_GridCompSetServices(), to use the functionality of the "producer" component through these three standard interfaces.

However, to resolve the second, third, and fourth issues, a data registration service has to be developed to match the variables of the "consumer" component with the variables of the "producer" component. This is a "consumer-producer" relationship, which has been coped with by the Common Component Architecture (CCA) in a generic way [13, 14, 15]. CCA uses "Use-Port" and "Provide-Port" to establish a generic interface between a consumer component and a producer component and hides the implementation details. CCA's approach will be used to guide the development of the generic part of our generic coupler.

With the data registration, the fourth issue can be resolved by building the required ESMF data structures: ESMF_State for the inter-component data exchange and ESMF_Field for regridding. The consistency of a user-defined variable name is enforced in the process of registration to coupling (regridding).

3. A generic coupler

As we discussed previously, a generic coupler for Earth system models needs to have a general and flexible data registration. In the following, we review portion of the data registration of the UCLA Distributed Data Broker (DDB).

The DDB is a software tool for coupling multiple, possibly heterogeneous, parallel models. One of the outstanding characteristics of the DDB is its data registration ability, at initialization time, to reconcile all requests ("consume") and offers ("produce") for data exchange between components.

There are two phases of registration. Each producer model component sends the "registration broker" an "offer" message, and each consumer model component sends a "request" message. (Of course, any process may simultaneously produce some quantities and consume others in a concurrent-run mode.) Once the "registration broker" has received all offers and requests, it starts the second phase in which it creates the lists of the matched offer-request pairs, forwards them to relevant producer model components, and notifies each consumer model component that each request will be satisfied by a certain number of data tiles.

Since our generic coupler is aimed at coupling models from different groups or organizations, a standard or well-known name convention should be used to facilitate matching names of exchanged variables from different models. The NetCDF Climate and Forecast Metadata Convention, the so-called CF Convention, is the most popular in the Earth system model community [16]. Thus, we will use it for illustration of our generic coupler. Of course, other name conventions can be used in our generic coupler.

3.1. Approach

Our generic coupler provides two services: 1) data registration and 2) data transfer and regridding. The data registration service will be derived from the DDB since its consumer-producer registration scheme is flexible and dynamic (real-time). However, the service of data transfer and regridding will be based on ESMF. That is because the existing DDB only supports rectilinear grids and the ESMF infrastructure is capable of supporting other grids and functionality.

Our generic coupler is designed to enable coupling of ESMF-compatible model components in an easy and general way. (An ESMF-compatible model has three standard interfaces, ESMF_GridCompInitialize(), ESMF_GridCompRun(), and ESMF_GridCompFinalize() and has a list of import and export variables for coupling with other models. But an ESMF-compliant model needs to use ESMF import/export states to exchange data in addition to having three standard interfaces.) The easy-to-use feature is implemented through relieving a user from manually building ESMF_Field and ESMF_State and coding the coupler. A user still deals with his/her familiar data type, namely Fortran arrays. Moreover, the general (abstraction) feature is implemented by adopting a well-known or standard name convention such as CF convention. The usage of a standard data variable name is similar to the usage of three standard interfaces mentioned above for functions.

3.2. Data registration service

Essentially each model component provides two lists of variable names: one for import ("consume") and another for export ("produce"). In each list, the model component needs to provide its user-defined name along with the corresponding CF name for import variables as well as export variables. The usage of the user-defined name is for the diagnostic purpose and optional. In addition to the CF name, the pointer to the grid metadata (i.e., starting and ending value as well as interval) is also needed for our generic coupler to build ESMF_field and then performing regridding operation between two coupled components.

The data registration service of our generic coupler takes these two kinds of lists, maps the import variable name with the corresponding export name based on their corresponding CF name, and provides the matched name list to the relevant consumer and producer model components.

Specifically, the following steps are taken consecutively:

1. Provide the variable name of exchanged data. Each component notifies the generic coupler of its variable requests, offers to supply, and any other required information (e.g., component logical name). For a concurrent-run mode, it is efficient for the generic coupler to use one computer node, while for a sequential-run mode the generic coupler uses all the nodes.

2. Match the name of an exchanged variable pair. The generic coupler (a) collects the messages of request and offer from all the components; (b) matches the request with the corresponding offer message; (c) builds the lists of "produce" and "consume" for each pair of the coupled components; (d) aggregates all the lists of "produce" and "consume" for each component. (Here we choose one import/export state for each component, that is, this import/export state is the aggregate of all the import/export variables from/to components. We believe that this aggregated approach will lead to a simple user interface); (e) distributes these lists and data structures to all the nodes in a sequential-run mode and relevant components in a concurrent-run mode.
3. Process the grid metadata and prepare regridding. The generic coupler (a) queries the grid meta data of each variable to be exchanged; (b) creates ESMF fields, import, and export states; (c) creates the data structures to hold information needed for regridding (e.g., pointers to import/export state, consumer/producer gridded component, ESMF VM). In particular, an ESMF regridding handle is created.

3.3. Data transfer and regridding

There are a few steps to be performed by our generic coupler during the run time. The three major steps are as follows:

1. Select the appropriate list to perform the requested data exchange. Specifically, the generic coupler determines which direction the data is to flow and which component pair will be performing the data transfer.
2. Call the ESMF_StateReconcile() subroutine to synchronize the data among the coupled component in a concurrent-run mode.
3. Call a component of our generic coupler to perform the inter-component data exchange as well as the regridding operations. A component of our generic coupler is developed to enable the functionality of our generic coupler to fit into the regular ESMF running flow and shift the flow from a gridded component to the generic coupler.

3.4. Run sequence of a generic coupler within a user driver

To use the generic coupler in a coupled system, a user will call the subroutines or utilities of ESMF gridded components as usual. But he or she will not call any subroutines or utilities of an ESMF coupler. Instead, a simple set of API's of the generic coupler is called directly. Specifically,

1. In the phase of component registration (e.g., call ESMF SetService subroutines), no call related to the coupler is needed.
2. In the phase of initialization (e.g., call ESMF Initialize subroutines), call the data registration subroutines of the generic coupler. Inside that registration subroutines, the subroutine of ESMF setService for the coupler component of

the generic coupler is called and the component data structure is set up (e.g., call ESMF_CplCompCreate()). At the end of the registration function, the subroutine of ESMF_CplCompIntialize() is called to complete the data registration and component initialization.

3. In the phase of Run (e.g., call ESMF Run subroutines), call the "doCouple" subroutines of the generic coupler to perform data transfer and regridding. Inside "doCouple," the subroutine of ESMF_CplCompRun() is called.
4. In the phase of Finalization (e.g., call ESMF Finalize subroutines), call the "finalizeCouple" subroutines of the generic coupler's to clean up. Inside "finalizeCouple," the subroutine of ESMF_CplCompFinalize() is called.

4. Discussion

So far, we have described the coupling issues and our solutions mostly for a sequential-run mode, although we also discussed the concurrent-run mode in some places. This is partly because ESMF has not completed its development for a concurrent-run mode. We believe that the data registration service of our generic coupler is applicable for the sequential- as well as the concurrent-run mode since it is derived from the DDB, which has been proven to be applicable for both modes. In the following, we clarify a few points for these two modes:

1. In the sequential-run mode, the generic coupler is operated in the combined set of computer nodes associated with all the model components. That is to ensure all the nodes are available to the generic coupler. This is the mode that the current ESMF is supporting.
2. In the concurrent-run mode, we envision the generic coupler will operate in the joint set of computer nodes, that is, the combined nodes of coupling model component pairs. For example, the joint set of nodes is 18 for the case in which model A uses 10 nodes and model B uses 8. The point is that the generic coupler is only operated for the component pair, not necessarily involving all the other components. In this case, the generic coupler between model A and model B will use one node to collect the name list from components, match names, create a name-matched list, and distribute the appropriate list to each gridded component, which is similar to the paradigm used by the DDB. When a component receives such a list, it can perform data transfer and regridding after calling ESMF_StateReconcile() subroutine to synchronize the data for regridding.

A software utility layer between ESMF and model components, called "Geo-generic", is being developed [17]. Its goal is also to simplify the usage of ESMF. However, our generic coupler focuses on the generic part of coupling through a data registration service. In addition, our generic coupler is applicable to sequential as well as concurrent-run modes.

5. Summary

We find that it is possible to build a generic coupler to simplify the usage of ESMF in coupling Earth system models. We have described our approach and discussed the implementation details based on CCA, DDB, and ESMF.

6. Acknowledgement

This project is supported by the NASA Earth-Sun System Technology Office Computational Technologies Project. We would like to thank C. Roberto Mechoso for helpful discussions and Tom Clune and Jarrett Cohen for reading the manuscript.

References

1. UCLA Distributed Data Broker, www.atmos.ucla.edu/~mechoso/esm/ddb_pp.html.
2. Goddard Earth Modeling System, unpublished
3. Flexible Modeling System, http://www.gfdl.noaa.gov/~fms
4. NCAR Flux Coupler, http://www.ccsm.ucar.edu/models/cpl/
5. ESMF, http://www.esmf.ucar.edu
6. C. Hill, C. DeLuca, V. Balaji, M. Suarez, A. da Silva, and the ESMF Joint Specification Team, "The Architecture of the Earth System Modeling Framework," Computing in Science and Engineering, Volume 6, Number 1, 2004.
7. S. Zhou et al., "Coupling Weather and Climate Models with the Earth System Modeling Framework," Concurrency Computation: Practice and Experience, to be submitted.
8. UCLA atmosphere model, http://www.atmos.ucla.edu/~mechoso/esm/
9. LANL POP ocean model, http://climate.lanl.gov/Models/POP/
10. MITgcm ocean model, http://mitgcm.org
11. fvCAM atmosphere model, http://www.ccsm.ucar.edu/models/atm-cam/
12. M. Kanamitsu, "Description of the NMC global data assimilation and forecast system," Wea. and Forecasting, Volume 4, 1989, pp. 335-342.
13. Common Component Architecture, http://www.cca-forum.org/
14. S. Zhou et al., "Prototyping of the ESMF Using DOE's CCA," NASA Earth Science Technology Conference 2003.
15. S. Zhou, "Coupling Climate Models with Earth System Modeling Framework and Common Component Architecture," Concurrency Computation: Practice and Experience, in press.
16. CF convections, http://www.cgd.ucar.edu/cms/eaton/cf-metadata/CF-20010808.html
17. Geo-generic, Max Suarez and Atanas Trayanov, private communication.