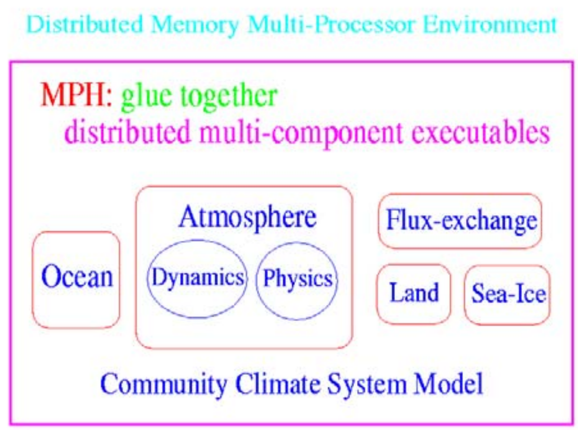
MPH: a Library for Coupling Multi-Component Climate Models and its Applications

Yun He and Chris H.Q. Ding, Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. yhe@lbl.gov, chqding@lbl.gov



A growing trend in developing large and complex applications on today's Teraflop scale computers is to integrate stand-alone and/or semi-independent program components developed by different institutions into a comprehensive simulation package. One example is the Community Climate System Model, which consists of atmosphere, ocean, land-surface and sea-ice components.

In CCSM, each component model is compiled into a binary executable image, like a standalone program. When CCSM is launched on a distributed memory computer, these component models are loaded into different set of processors; however, each model does not know the other models, their names, their processor-allocations, etc. A critical task on such distributed memory multi-component computer is a handshaking process that enables every component models to get the necessary information about other component models.

We develop a general-purpose multi-program handshaking (MPH) library to setup the distributed multi-component environment: for different executables to recognize each other, for setting up a registry of executable names and communication channels among different components. MPH must meet the following requirements: (a) Flexible component names. As CCSM is developed, component model and their names evolve. For example, atmosphere model changes from CCM to CAM, land-surface model changes from LSM to CLM, etc. Thus component names cannot be hardwired into the coupler. (b) Allow several model integration mechanisms. In CCSM 1, each component model is a single executable. In the related PCM (parallel climate model), each component model is a subroutine, and all component models are compiled into a single executable. As CCSM evolves, a component model could have several sub-components. Final ensemble simulations requires yet another multi-instance mechanism. (c) Resource allocation. Processor allocation must be flexible and only need to be specified at runtime through a simple controlling mechanism. All these requirements are met by MPH. In addition, a number of further utilities are provided as well.

For the first time, we clearly identify five effective execution modes and develop the MPH library to support application developments for utilizing these modes.

(1) Single-Component Executable, Single-Executable application (SCSE)
(2) Single-Component Executable, Multi-Executable application (SCME) **CCSM mode**
(3) Multi-Component Executable, Single-Executable application (MCSE) **PCM mode**
(4) Multi-Component Executable, Multi-Executable application (MCME) **most flexible mode**
(5) Multi-Instance Executable, Multi-Executable application (MIME) **ensemble simulations**

MPH provides the key infrastructure for integrating separate executables together. It provides functionalities for component name registration, resource allocation and initialize communication channels between independent components. MPH also supports components-joining, inter-component communication, inquiry on multi-component environment, and redirect input/output. MPH provides a flexible, versatile mechanism for these tasks, which are foundations for larger software tools/frameworks.

MPH provides a convenient framework to do the ensemble simulations. A multi-instance executable is a special type of executable. It differs from regular single-component and multi-component executables in that this particular executable is replicated multiple times (multiple instances) on different processor subsets. This enables running ensembles simultaneously as a single job, and ensemble averaging being done on the fly. Not only is this an effective way of using existing computing resources, it also reduces the potential human error by reducing the number of jobs that need to monitored. This eliminates large data output and storage for post-processing averaging, and enables on-the-fly nonlinear ensemble statistics that are otherwise impossible to compute as a post-processing step.

All MPH functionalities are currently working on IBM SP, SGI Origin, HP AlphaServer SC, and Linux clusters. MPH has been adopted in CCSM development, which is the U.S. flagship coupled climate model system heavily used in long-term climate research and government policy matters. A Model Coupling Toolkit for communication between different component models uses MPH. MPH has also been adopted in NCAR's Weather Research and Forecast (WRF) model and the Colorado State University's geodesic grid coupled model. Edinburgh Parallel Computing Centre (EPCC) uses MPH for ensemble simulations. Many other users for multi-instance simulations include MM5, WRF, ECMWF, Ocean DieCAST model, and a Monte Carlo code running on 1024 processors.

One of the applications for MPH is the concurrent single-executable development of CCSM. A multi-executable code integrates these components together as a single computational system while keep each component as a standalone executable. CCSM is currently such a multi-executable system based on the MPMD mechanism. It is cumbersome in usage and not available for machines without MPMD. So, single-executable CCSM is under request. We are developing a concurrent single-executable version of CCSM that coexists with multi-executable option. It is accomplished by redesigning the top level CCSM structures using MPH. We also proposed a module-based approach to solve name conflict issues associated with single-executable CCSM.

Both MPH multi-instances and single-executable development work improve future HEC capability for realizing next generation climate models. Detailed information is at: http://hpcrd.lbl.gov/SCG/acpi/MPH/ and http://hpcrd.lbl.gov/SCG/acpi/SE.

Acknowledgement: MPH is developed in collaboration with Tony Craig, Brian Kauffman, Vince Wayland and Tom Bettge of National Center of Atmospheric Research, and Rob Jacobs and Jay Larson of Argonne National Laboratory. This work is supported by a DOE SciDAC Climate Project and NERSC Program.

References:

Y. He and C. Ding, Coupling Multi-Component Models with MPH on Distributed Memory Computer Architectures, *International Journal of High Performance Computing Applications*, Vol.19, No.3, 329-340, August 2005. Earlier versions in *Proceedings of International Parallel and Distributed Processing Symposium*, April 2004, and *Proceedings of Realizing Teracomputing, the Tenth ECMWF Workshop on the Use of High Performance Computing in Meteorology,* November 2002.