# Making a single executable version of WRF 4Dvar with ESMF

Donald Stark[1], Xiang-Yu Huang, and Xin Zhang

National Center for Atmospheric Research, Boulder, Colorado

## 1. Introduction

The Weather Research and Forecasting Data Assimilation system (WRFDA), developed at NCAR/MMM, is a unified WRF-based (global/regional, multi-model, 3/4DVAR) model-space variational data assimilation system (*Huang et al. 2009; Huang et al. 2005*). In terms of software, the 4DVAR system consists of three concurrently running executables that communicate through disk I/O and UNIX system calls. The use of disk I/O for inter-model communication has the advantage of reducing memory requirements, with the trade off being reduced performance. The three executables are, **WRF VAR** - the variational code, **WRF NL** - the forward nonlinear model, and **WRF PLUS** - the reduced physics tangent linear and corresponding adjoint models.

The WRF 4DVAR system employs the incremental 4D-Var formulation commonly used in operational systems (*Courtier et al. 1994; Barker et al. 2006*). The incremental approach minimizes a cost function defined in terms of the analysis increment, rather than the analysis itself. The incremental 4D-Var solver employs a double loop structure, consisting of an inner and outer loop. The outer loop deals with the nonlinear aspects of the assimilation, while the inner loop conducts the minimization of a quadratic cost function.

The WRF 4DVAR system makes use of many previously developed components of the WRF 3DVAR system (*Barker et al. 2005*), observation operators, quality control, and the background error covariance model. In addition it employs a minimization inner-loop using simplified WRF tangent linear and adjoint models, assumes Gaussian error covariances, and an iterative outer loop using the nonlinear WRF model to update the basic trajectory state to account for the effect of nonlinearities in the[1] assimilation algorithm.

While the current WRFDA data assimilation system has been a useful research tool for studying the impact of data assimilation in the weather community, it is hampered by the complexity of the multiple executable system and the performance costs associated with the disk based communication scheme. To remedy this, a single executable version of the WRF 4DVAR system is being created using the Earth System Modeling Framework (ESMF) to transform the multiple executable, concurrent system, into a coupled single executable system, where communication occurs through an MPI based coupler. The benefit of switching from the multiple executable concurrent system, to a hybrid concurrent-sequential design, is the potential for increased performance by reducing processor idle time, with the trade off being increased internal memory usage.

This paper will first discuss the issues of adopting ESMF as a system framework, and then move on to examine the standard, and not so standard, coupling methods available by using ESMF. Next, will be the deconstruction of the WRF 4DVAR solver into ESMF coupled components, followed by a discussion of the design and implementation of an ESMF coupling system for the WRF 4DVAR, highlighting the implementation challenges faced.

## 2. ESMF

ESMF (http://www.esmf.ucar.edu/), developed at NCAR, is a community based model coupling framework, intended to create a flexible, high-performance, software infrastructure that facilitates the performance, portability, interoperability, and reuse of Earth science code. ESMF defines an architecture for composing complex, coupled modeling systems and includes data structures and utilities for developing individual models.

The basic idea behind ESMF is that complicated applications should be broken up into smaller pieces that ESMF calls components. Formally, a component is a unit of software composition that has a coherent function, with a standard calling interface and behavior. Standard interfaces help to foster interoperability of components, and the reuse of components in different contexts. Component-based design is often a natural fit for Earth system models since typically components are comprised of a set of substantial, distinct and loosely interacting domains, such as atmosphere, land, sea ice and ocean. Components can also be comprised of separable parts within a model,

[1] Primary contact: stark@ucar.edu

such as the dynamics, physics, and chemistry. Care must be taken to carefully design such tightly coupled systems of components in order to not adversely impact performance.

Coupler components can be written to transform data between a pair of components, or a single coupler component can couple multiple components. Multiple couplers may be included in a single modeling application. This is a natural strategy when the application is structured as a hierarchy of components. Each level in the hierarchy usually has its own set of coupler components.

All data exchanged between components is stored in ESMF import and export State objects. These are simple containers that hold native ESMF data types.

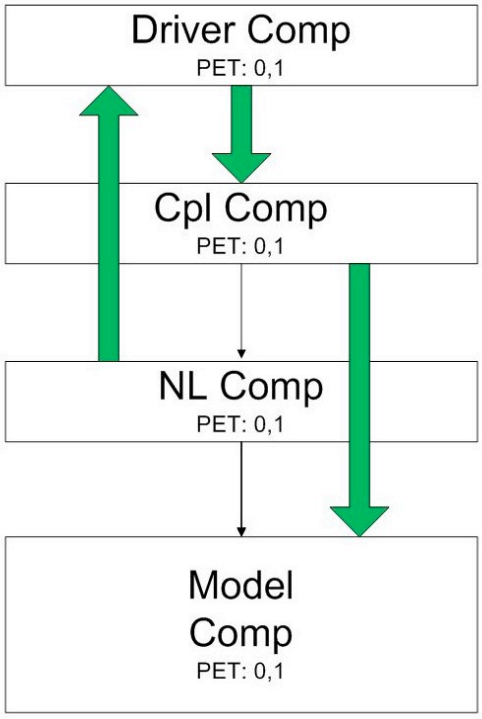


Figure 1. Hub and spoke inter-component data exchange.

A very simple ESMF application might involve an application driver or parent component, two child components that require inter-component data exchange, and two coupler components. One coupler component moves data forward, and the other moves data back.

Typically, an inter-component data exchange starts with the run phase of the first child component being interrupted. Program control is then returned to the driver, which in turn activates the coupler. The coupler receives an export state from the first child component and transfers it to the import state of the second child state. Program control then returns to the driver. This sort of coupling is commonly referred to as hub and spoke. (See figure 1.) It assumes that coupled components are loosely linked and that the linkage is easily separable.

A completely different strategy for inter-component data exchange is Direct coupling. Direct coupling is a method to initiate a data exchange without needing to first return to a coupler component interface. The data exchange is arranged within a coupler component, usually at initialization time, but it can be invoked from deep within a child component. Direct coupling assumes that the two child components are running concurrently. This is useful when modeling tightly linked physical processes. (See figure 2.)

There is no single, generic Coupler Component for all ESMF applications. Form is strongly dependent on function.

Modelers write Coupler Component internals using ESMF classes bundled with the framework. These classes include, among other things, methods for time advancement, data redistribution, calculation of interpolation weights, interpolation, and subscription of processor resources.

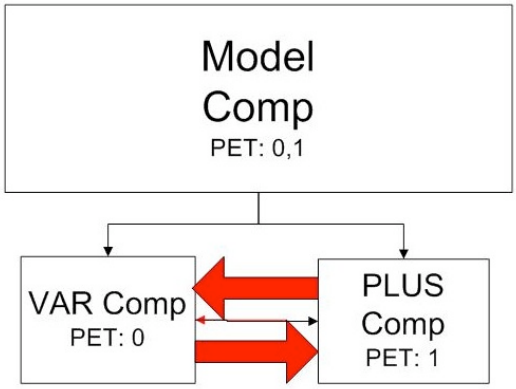


Figure 2. Direct coupling inter-component data exchange. This method assumes the two child components are concurrently running processes.

The steps used in adopting ESMF can be summarized by the acronym **PARSE**:

i). **P**repare user code. Split user code into initialize, run, and finalize methods and decide on components, coupling fields, and control flow.
ii). **A**dapt data structures. Wrap native model data structures in ESMF data structures to conform to ESMF interfaces.

iii). **R**egister user methods. Attach user code initialize, run, and finalize methods to ESMF components through registration calls.
iv). **S**chedule, synchronize, and send data between components. Write couplers using ESMF redistribution, sparse matrix multiply, regridding, and/or user-specified transformations.
v). **E**xecute the application. Run components using an ESMF driver.

### 3. Multiple executable WRF 4DVAR

The execution of the current WRF 4DVAR system begins by launching a sequence of scripts, which set up the run time environment, construct the input namelist files, and create the hierarchy of working directories. Once the set up is completed, the three component executables, **WRF VAR**, **WRF NL**, and **WRF PLUS** are launched. The three executables run concurrently. Each separately initializes itself, digests input files and then moves on to the run phase of the analysis.

Once in the run phase, both the **WRF NL** and the **WRF PLUS** executables enter a loop where they pause until signaled, through UNIX system calls, by the **WRF VAR** executable, to continue. The **WRF VAR** executable proceeds into the data assimilation solver routine (called ***da_solve***), which contains the inner and outer loops of the solver. **WRF VAR** advances the outer loop by initiating a system call, which signals the nonlinear forward model (**WRF NL**) to propagate the background state to the end of the analysis window. While **WRF NL** is running, **WRF VAR** enters a loop where it waits for a system call to tell it that **WRF NL** has completed. Once **WRF NL** has completed, it goes into a standby mode, waiting for another outer loop iteration. **WRF VAR** then continues its execution and enters the inner loop.

The inner loop solves for the analysis by minimizing the incremental cost function. It does this by employing a Conjugate Gradient solver. The minimization functions by iterating the solution with the tangent linear and then the adjoint model. For a single outer loop iteration, the inner loop iterates on the order of a hundred times between the **WRF VAR** executable and the **WRF PLUS** executable. Once the cost function is minimized, **WRF PLUS** goes into standby mode and waits for another outer loop. The **WRF VAR** regains control of the execution and produces the analysis field.

### 4a. WRF 4DVAR Redesign – Outer loop

The traditional approach to coupling design in ESMF can be best described as hub and spoke. Two completely separable components are connected together by a third component, which acts as a coupler. In figure 1, the first component (NL Comp) completes a stage of its run phase and returns control to the Driver Comp (upward arrow). The Driver Comp activates the coupler (Cpl Comp), (downward arrow from the driver to the coupler), which transfers a state from NL Comp to the Model Comp (downward arrow from Cpl Comp to Model Comp) through the Cpl Comp. When the transfer is completed, control is returned to the Driver Comp.

It is important to emphasize that with hub and spoke coupling, it is necessary to interrupt the run phase of a component, return to the main driver, and then from the main driver, initiate the coupler. This requires that the run phase of the component must be separable into multiple phases. As we will see for the inner loop, this is not always the optimal choice.

The hub and spoke coupling paradigm, is an ideal solution for coupling the nonlinear model (**WRF NL**) with the variational code (**WRF VAR**). It only requires unwinding the outer loop in ***da_solve***, and constructing two new routines, call them ***da_solve_nl*** for the call to the nonlinear model, and ***da_solve_minJ*** for the minimization of the cost function. In the new code structure, it is also necessary to pull the outer loop up into to the driver component. In pseudo code it would look roughly like the following:

```
! in main driver
do iterate outer loop
      call da_solve_nl()
      call da_solve_minJ()
end do outer loop
```

The hub and spoke approach satisfies the coupling requirements for the outer loop since the call to the nonlinear model can be easily separated from the original routine, leaving only the solver to minimize the cost function.

### 4b. WRF 4DVAR Redesign – Inner loop

While the outer loop coupling is amenable to a traditional hub and spoke strategy, the inner loop is not. Due to the strong linkage between the **WRF VAR** and **WRF PLUS** components within the inner loop, as well as the large number of iterations between the two components to minimize the cost function, separating the loop into two phases would be difficult and inefficient. A better approach is the ESMF Direct coupling.

Direct coupling works by directly connecting the source and destination through use of separate one sided communication calls. The direct coupling avoids the need to interrupt the current run method to do an exchange. The addresses for the direct communication are precomputed during the initialization phase and sent to the run method using a standard hub and spoke style of coupling.

The direct coupling method is represented in figure 2. The two components, VAR Comp and Plus Comp, are directly linked (red arrows) without interrupting the current run phase. There is no need to return program control to the main driver before calling the coupler. Instead, the two components run concurrently. When each reaches the point in its run phase where it needs to couple, a pair of *put* and *get* calls are initiated. The execution is blocked until the communication completed, when the run phase continues uninterrupted.

**4c. WRF 4DVAR Redesign – overview**

The combined single executable WRF 4DVAR system is represented in figure 3. The Driver Comp activates the nonlinear forward model NL the Driver Comp. The Driver Comp activates the Comp. It runs to completion and returns control to Cpl Comp. The Cpl Comp transfers the trajectories generated in the NL Comp to the Model Comp. Model Comp shares these with the two components VAR Comp and Plus Comp and starts the concurrent execution of the two components.

This part of the operation is iterated to minimize the cost function. The VAR Comp communicates directly with the tangent linear solver in the Plus Comp (Plus-TL), sends it the initial state and tells it to integrate tangent linear forward in time. The Plus-TL releases its control sends the tangent linear model states back to the VAR Comp and moves on to the adjoint part of the Plus Comp (Plus-AD). From the tangent linear model states, the VAR Comp computes the adjoint forcing and then communicates directly with the Plus-AD. The Plus-AD integrates the adjoint backwards and returns adjoint model integration and control back to the VAR Comp. From here, if the convergence criteria are not met, the process repeats.

**5. Conclusions**

This hybrid sequential-concurrent design purposely employs two very different communication strategies, hub and spoke for the coupling between the nonlinear code and the variational code, and direct communication between the variational and the tangent linear/adjoint code. This approach is intended to balance between generality and performance.

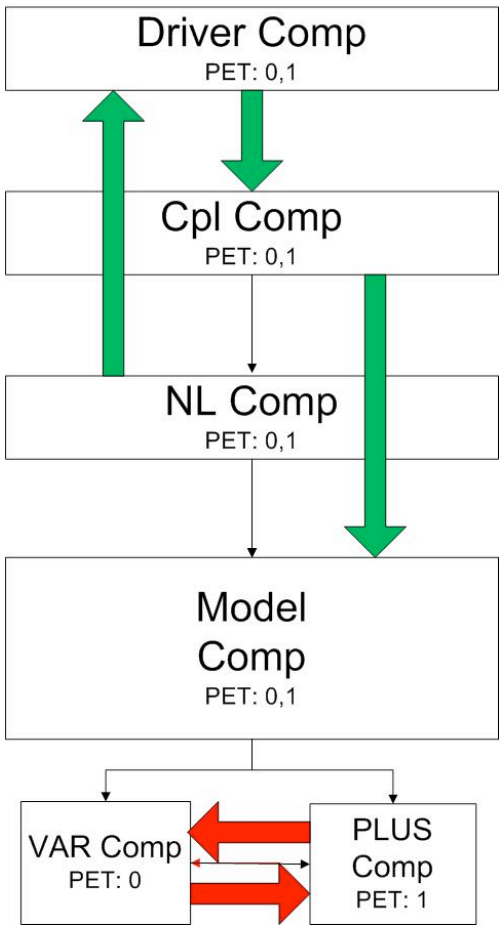


Figure 3. Component diagram of the single executable WRF 4DVAR.

The hub and spoke coupling is the most general approach, requiring the least work to swap components, and working for both a sequential and concurrent design. The direct coupling is the least general, being exclusively set up for specific versions of the WRF tangent linear and adjoint codes. The direct coupling also requires that the coupled components be running concurrently. Where direct coupling has the advantage, is when the two components being coupled are so tightly linked that separation is not practical. This is certainly the case with the coupling between the VAR and PLUS components.

All things being equal, the performance of the two strategies is going to be the same. Although direct coupling has the potential, if not carefully load balanced, to be less efficient due to the possibility of one of the concurrent processes to sit idle and waste CPU cycles while waiting for the other.

## 5. Current Status

The coupled single executable system is still under development. The superstructure, driver, coupler, and model components are complete. The wrapper for the nonlinear model (**WRF NL**) and the variational component (**WRF VAR**) are complete. The direct coupling calls inside of the **WRF PLUS** are currently being implemented, and run time bugs are still being debugged.

## References


**Barker, D.M., J. Bray, Y.-R. Guo, X.-Y. Huang, Z. Liu, S. Rizvi, Q.-N. Xiao**, 2006: Status Report on WRF-ARW's Variational Data Assimilation System (WRF-Var). WRF users' workshop, Boulder, Colorado, 19-22 June 2006.

**Courtier, P., J.-N. The'paut, A. Hollingsworth**, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart. J. Roy. Meteor. Soc.* **120**, 1367-1387.

**Huang, X.-Y., Q. Xiao, W. Huang, D.M. Barker, Y.-H. Kuo, J. Michalakes, Z. Ma**, 2005: The Weather Research and Forecasting Model Based 4-dimensional Variational Data Assimilation System. WRF users' workshop, Boulder, Colorado, June 2005.

**Huang, X.-Y., et al.**, 2009: Four-Dimensional Variational Data Assimilation for WRF: Formulation and preliminary Results. *Mon. Wea. Rev.* , **137**, 299-313.