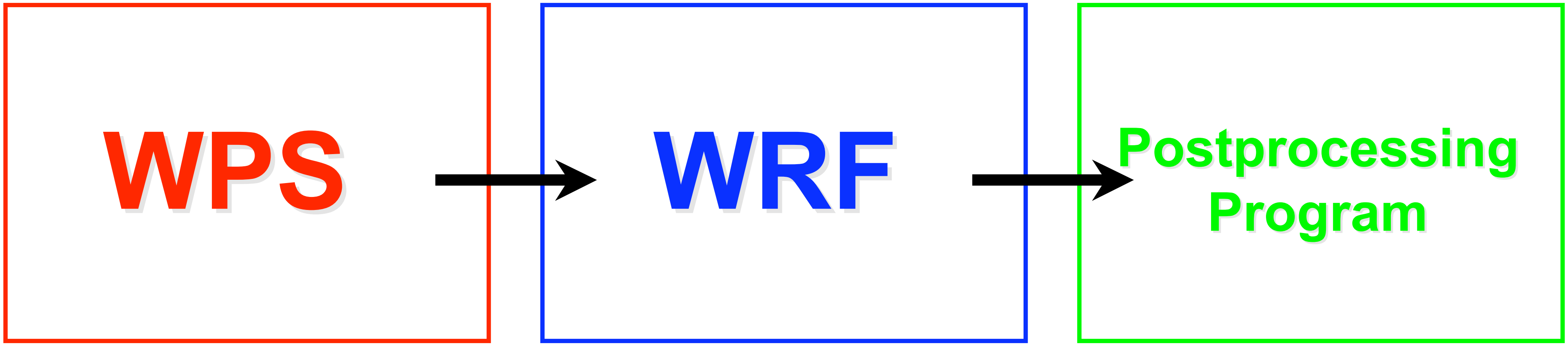
WRF and WPS: Compile

Jamie Wolff
National Center for Atmospheric Research (NCAR)
The Developmental Testbed Center (DTC)

24th July, 2007

Program Flow



WPS

WRF

Postprocessing Program

- Share common routines such as WRF I/O API

*First need to compile WRF and then WPS

System Requirements

Required libraries (WRF and WPS):

- FORTRAN 90/95 compiler
- C compiler
- Perl
- netCDF
- MPI (required regardless of the number of processors for NMM)
- NCAR Graphics (optional, but recommended – used by graphical utility programs)

Optional libraries* for GRIB2 support (WPS):

- JasPer (JPEG 2000 “lossy” compression library)
- PNG (“lossless” compression library)
- zlib (compression library used by PNG

*Installation of these libraries in not part of the WPS installation script

We recommend having a system administrator install the required libraries before installing WRF or WPS

Installing WRF

- Download source code
- Set environment
- Configure and Compile WRF

Download WRF Source Code

- The WRF source code can be obtained from: *http://www.mmm.ucar.edu/wrf/users/downloads.html*
  - Click ‘WRF’ on the side menu, then
    - ‘New Users’, register and download, or
    - ‘Returning User’, enter email and download
- Both the ARW and NMM cores are included in:

  WRFV2.2.TAR.gz *(or the latest release available)*

NOTE: *A newer version of the WRF code (which supports WPS output for the NMM) will be used for this tutorial*

- After ***gunzip*** and ***untar***, should see a directory **WRFV2/**

  tar –zxvf WRFV2.2.TAR.gz
- ***cd*** to **WRFV2/** directory

WRFV2 Directory

	Makefile	Top-level makefile
	README	General information about WRF code
	README.NMM	NMM specific information
	README test cases	Explan ation of the test cases for ARW
Data dictionary	Registry/	Directory for WRF Registry file
Compile rules	arch/	Directory where compile options are gathe red
	chem/	Directory for chemistry modules
Compile scripts	clean	script to clean created files and executables
	compile	script for compiling WRF code
	configure	script to configure the configure.wrf file for compile
	dyn_em	Directory for ARW dynamic modules
	dyn_exp/	Directory for a 'toy' dynamic core
	dyn_nmm/	Directory for NMM dynamic modules
Source code directories	external/	Directory that contains external packages, such as those for IO, time keeping and MPI
	frame/	Directory that contains modules for WRF framework
	inc/	Directory that con tains include files
	main/	Directory for main routines, such as wrf.F, and all executables
	phys/	Directory for all physics modules
Run directories	run/	Directory where one may run WRF
	share/	Directory that contains mostly modules for WRF mediation layer and WRF I/O
	test/	Directory containing sub -directories where one may run specific configurations of WRF.
	tools/	Directory that contains tools

Set environment

- If the ***netCDF*** is not in the standard ***/usr/local*** then set the **NETCDF** environment variable before typing ‘***./configure***’:

  Example: ***setenv NETCDF /usr/local/netcdf-pgi***

- As a general rule for LINUX systems, make sure the ***netCDF*** and ***MPI*** libraries are installed using the same compiler (PGI, Intel, g95) that will be used to compile WRF.

Configuring WRF

- To create a WRF configuration file for your computer, type:

  ./configure

- This script checks the system hardware and software (mostly *netCDF*), and then offers the user choices for compiling WRF:

  - Serial, OpenMP, or MPI
  - RSL or RSL_LITE (interface to MPI)
  - Nesting or no nesting

List of Configure Options - I

Choices for 32-bit LINUX operated machines are:

1. PC Linux i486 i586 i686, PGI compiler (Single-threaded, no nesting)
2. PC Linux i486 i586 i686, PGI compiler (Single threaded, allows ARW nesting using RSL without MPI)
3. PC Linux i486 i586 i686, PGI compiler SM-Parallel (OpenMP, no nesting)
4. PC Linux i486 i586 i686, PGI compiler SM-Parallel (OpenMP, allows ARW nesting using RSL without MPI)
5. **PC Linux i486 i586 i686, PGI compiler DM-Parallel (RSL, MPICH, allows ARW nesting)**
6. **PC Linux i486 i586 i686, PGI compiler DM-Parallel (RSL_LITE, MPICH, allows ARW and NMM nesting)**
7. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (Single-threaded, no nesting)
8. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (Single threaded, allows ARW nesting using RSL without MPI)
9. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler (OpenMP)
10. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort compiler SM-Parallel (OpenMP, supports ARW nesting using RSL without MPI)
11. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+icc compiler DM-Parallel (RSL, MPICH, with ARW nesting)
12. AMD x86_64 Intel xeon i686 ia32 Xeon Linux, ifort+gcc compiler DM-Parallel (RSL, MPICH, with ARW nesting)
13. PC Linux i486 i586 i686, g95 compiler (Single-threaded, no nesting)
14. PC Linux i486 i586 i686, g95 compiler DM-Parallel (RSL_LITE, MPICH, allows ARW and NMM nesting)

ARW: all options are available

NMM: must have MPICH (options 5, 6*, 11, 12, 14) and nesting will ONLY work with RSL_LITE

List of Configure Options - II

Choices for IBM machines are:

1. AIX (single-threaded, no nesting)
2. AIX SM (OpenMP, no nesting)
3. **AIX DM-Parallel (RSL_LITE, IBM-MPI, allows ARW and NMM nesting)**
4. **AIX DM-Parallel (RSL, IBM-MPI, allows ARW nesting)**
5. AIX DM-Parallel (RSL, IBM-MPI, allows ARW nesting) (PARALLEL HDF5)
6. AIX DM-Parallel (RSL_LITE, IBM-MPI, allows ARW and NMM nesting) (PARALLEL HDF5)
7. AIX DM-Parallel/SM-Parallel (not recommended) (RSL, IBM-MPI, OpenMP, allows ARW nesting)
8. AIX DM-Parallel (RSL, IBM-MPI, MCEL, experimental!)
9. AIX DM-Parallel ESMF (RSL, IBM-MPI, ESMF coupling, no nesting, experimental!)
10. AIX (Single-threaded, ARW nesting using RSL without MPI)
12. AIX (OpenMP, ARW nesting using RSL without MPI)

ARW: all options are available; 3* recommended

***NMM: must have MPI (options 3*, 4, 5, 6, 9)
and nesting will ONLY work with RSL_LITE***

Configuring WRF, cont.

- The *./configure* command will create a file called ***configure.wrf***

  This file contains compilation options, rules, etc. specific to your computer and can be edited to change compile options, if desired.

- WRFV2 compile options are provided for a number of platforms. In addition, the ***arch/configure.defaults*** file can be edited to add a new option if needed.

Compiling WRF

- First set ***one*** core environment variable to 1:

ARW: ***setenv WRF_EM_CORE 1***

NMM: ***setenv WRF_NMM_CORE 1***

Note: If neither of these environment variables are set, the default is to compile ARW.

In addition, if running NMM with nesting:

setenv WRF_NMM_NEST 1

Compiling WRF

- Type the following command to compile:

./compile test_case >& compile_wrf.log

where ***test_case*** is one of the following:

compile em_b_wave	*3D Ideal Case (ARW only)*
compile em_quarter_ss	
compile em_grav2d_x	*2D Ideal Case (ARW only)*
compile em_hill2d_x	
compile em_squall2d_x	
compile em_squall2d_y	
compile em_real	***Real Data Cases (ARW and NMM)***
compile nmm_real	

compile -h ***help message***

Compiling ARW: Idealized Cases

- If the choosen ideal case compilation is successful, it will create two executables:
  - ✓ ***ideal.exe***: used for ARW initialization of ideal cases.
  - ✓ ***wrf.exe***: used for ARW model integration.
- These executables will be linked to the specific **test/em_*test_case*** and **run** directories.

Compiling WRF: Real Data Case

- If the real data case compilation is successful:
  - ARW: creates four executables in the ***main/*** directory:
    - ✓***real.exe***: used for initialization of real data cases.
    - ✓***wrf.exe***: used for model integration.
    - ✓***ndown.exe:*** used for one-way nesting
  - NMM: creates two executables in the ***main/*** directory:
    - ✓***real_nmm.exe***: used for initialization of real data cases.
    - ✓***wrf.exe***: used for model integration.
- These executables will be linked to either ***test/em_real*** or ***test/nmm_real*** and ***run/*** directories.

Compiling both WRF cores

Core "A"

- Set environment
- Configure, compile
- Save ***main/wrf.exe*** to ***main/wrf_coreA.exe***
- Copy ***main/*exe*** to a temporary location outside of WRFV2/

clean -a

Core "B"

- Set environment
- Configure, compile
- Save ***wrf.exe*** to ***wrf_coreB.exe***

Move Core "A" ***exe's** from temporary location back to ***WRFV2/main*** (and to ***test/test_case/*** if you run there)

Installing WPS

- Download static terrestrial data
- Download source code
- Configure and Compile WPS

Reminder: A successful compilation of WRF is required prior to WPS compilation!

Download Static Terrestrial Data

- The terrestrial fields interpolated by ***geogrid*** may be downloaded from same page as the code:

  http://www.mmm.ucar.edu/wrf/users/download/get_source.html

  - Two options for data: low-res and all resolutions

- Data are static: only need to be downloaded once

Download Static Terrestrial Data, Cont.

- The *geog.tar.gz* file (all resolutions) contains:
  - *albedo_ncep* – monthly surface albedo
  - *greenfrac* – monthly vegetation fraction
  - *islope* – slope index
  - *landuse* – land use category (30”, 2’, 5’, and 10’ res.)
  - *maxsnowalb* – maximum snow albedo (30”, 2’, 5’, and 10’ res.)
  - *soiltemp* – annual mean deep soil temperature (30”, 2’, 5’, and 10’ res.)
  - *soiltype_bot* – bottom-layer soil type (30”, 2’, 5’, and 10’ res.)
  - *soiltype_top* – top-layer soil type (30”, 2’, 5’, and 10’ res.)
  - *topo* – topography height (30”, 2’, 5’, and 10’ res.)

Download Static Terrestrial Data, Cont.

- Uncompress the data into a directory with ~10 GB of available space (264 MB for low-res only)!

  ```
  tar -zxvf geog.tar.gz
  ```

- Data can be shared by users on the same machine by placing files in a common directory
  - Recommended due to size!

Download WPS Source Code

- The WPS source code can be obtained from:

  http://www.mmm.ucar.edu/wrf/users/downloads.html

- WPS is designed to work with WRF v2.2 for ARW and WRF v2.2.1 for NMM (or later)
  - *WPS programs use WRF I/O API libraries to do file input and output*
  - *These I/O libraries are built when WRF is installed*

NOTE: A newer version of the WPS code (which supports NMM) will be used for this tutorial

- For simplicity, install WPS/ in the same location as WRFV2/
- After ***gunzip*** and ***untar***, should see a directory WPS/

  tar –zxvf WPSV2.2.TAR.gz *(or the latest release available)*

  ls

  WPS/   WRFV2/
- ***cd*** to WPS/ directory

Configure WPS

- To create a WPS configuration file for your computer, type:

  ./configure

- This script offers the user choices for compiling WPS:

  - Serial or Distributed memory

  - GRIB1 or GRIB2

- The *./configure* command will create a file called ***configure.wps***

  This file contains compilation options, rules, etc. specific to your computer and can be edited to change compile options, if desired.

List of WPS Configure Options

```
Will use NETCDF in dir: /usr/local/netcdf-pgi
    $JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2
    I/O...
    ------------------------------------------------------------------------
Please select from among the following supported platforms.

       1.  PC Linux i486 i586 i686, PGI compiler          serial, NO GRIB2
       2.  PC Linux i486 i586 i686, PGI compiler          serial
       3.  PC Linux i486 i586 i686, PGI compiler          DM parallel, NO GRIB2
       4.  PC Linux i486 i586 i686, PGI compiler          DM parallel
       5.  PC Linux i486 i586 i686, Intel compiler        serial, NO GRIB2
       6.  PC Linux i486 i586 i686, Intel compiler        serial
       7.  PC Linux i486 i586 i686, Intel compiler        DM parallel, NO GRIB2
       8.  PC Linux i486 i586 i686, Intel compiler        DM parallel
       9.  PC Linux i486 i586 i686, g95 compiler,         serial, NO GRIB2
      10.  PC Linux i486 i586 i686, g95 compiler,         serial

    Enter selection [1-10] : 1
```

(***ARW* and *NMM***: all options available, *recommended to use serial, NO GRIB2)

```
----------------------------------------------------------
Configuration successful. To build the WPS, type: compile
----------------------------------------------------------
```

Compile WPS

- If configuration was successful, compile WPS:

```
./compile >& compile_wps.log
```

- If the compilation is successful, it will create three executables:
  - ✓ ***geogrid.exe***: define size/location of domain(s).
  - ✓ ***ungrib.exe***: extract meteorological fields from GRIB files.
  - ✓ ***metgrid.exe***: horizontally interpolate meteorological fields (from *ungrib*) to simulation grid(s) (defined by *geogrid*)

Compile WPS, Cont.

- If NCAR Graphics libraries are available and compilation is successful, it will also create seven executables in ***util/***:
  - ✓ ***avg_tsfc.exe***
  - ✓ ***g1print.exe***
  - ✓ ***g2print.exe***
  - ✓ ***mod_levs.exe***
  - ✓ ***plotgrids.exe***
  - ✓ ***plotfmt.exe***
  - ✓ ***rd_intermediate.exe***

  – Each of these utilities are described in more detail in the WPS Overview talk

Sharing WPS Installation

- A single build of WPS will work for both ARW and NMM core
- Multiple users may share a single installation of the WPS; not every user needs to install
  - WPS installation directory read-only
  - Each user will run WPS programs in their own working directories
  - Output files created in user working directories

Additional Resources

- For more detailed information on installation of WRF and WPS, please see:
  - ARW and NMM Users Guides
  - Online Users Pages:
    - **ARW:** *http://www.mmm.ucar.edu/wrf/users/*
    - **NMM:** *http://www.dtcenter.org/wrf-nmm/users/*
- For further assistance regarding WRF and WPS:
  - WRF Users Forum: *http://tornado.meso.com/wrf_forum/*
  - WRF Help email: *wrfhelp@ucar.edu*

Questions?