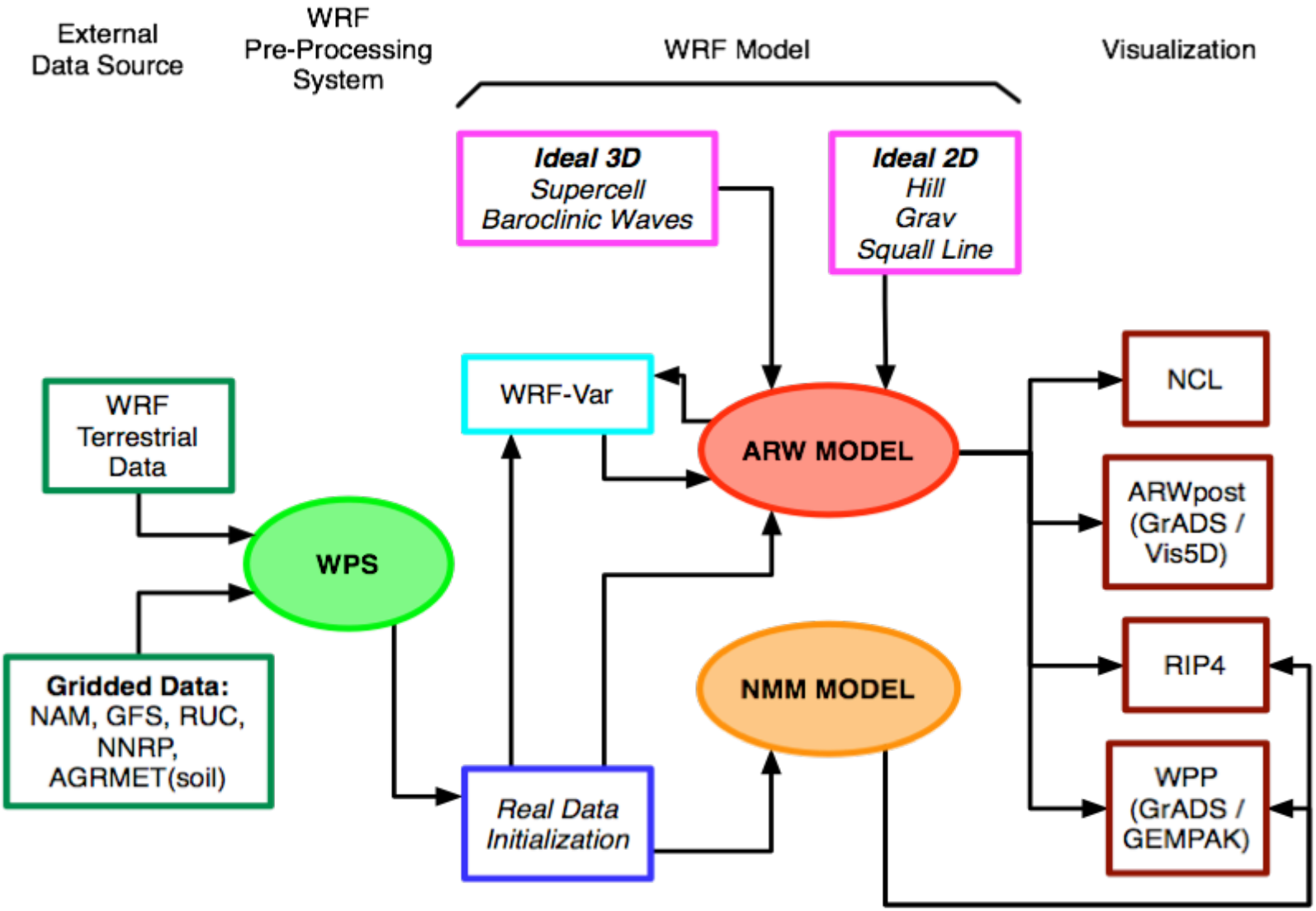
WRF and WPS: Compile

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

15th January, 2008

Program Flow



- WRF and WPS 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:

  WRFV2.2.1.TAR.gz *(or the latest release available)*
- After ***gunzip*** and ***untar***, should see a directory **WRFV2/**

  tar –zxvf WRFV2.2.1.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	Explanation of the test cases for ARW
Registry/	Directory for WRF Registry file
arch/	Directory where compile options are gathered
clean	script to clean created files and executables
compile	script for compiling WRF code
configure	script to configure the configure.wrf file for compile
chem/	Directory for chemistry modules
dyn_em	Directory for ARW dynamic modules
dyn_exp/	Directory for a 'toy' dynamic core
dyn_nmm/	Directory for NMM dynamic modules
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 contains include files
main/	Directory for main routines, such as wrf.F, and all executables
phys/	Directory for all physics modules
share/	Directory that contains mostly modules for WRF mediation layer and WRF I/O
tools/	Directory that contains tools
run/	Directory where one may run WRF
test/	Directory containing sub-directories where one may run specific configurations of WRF.

Data dictionary →

Compile rules →

Compile scripts →

Source code directories

Run directories →

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***

- WRF needs both the ***lib*** and ***include*** directories
- 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.

Set environment, cont.

- Most of these settings are not required, but if difficulties are encountered you may want to try:
  - ***unset limits***
    - Especially if you are on a small system
  - ***setenv MP_STACK_SIZE 64000000***
    - OpenMP blows through the stack size, set it large
  - ***setenv MPICH_F90 f90*** (Or whatever FORTRAN compiler may be called)
    - WRF needs the ***bin***, ***lib*** and ***include*** directories
  - ***setenv OMP_NUM_THREADS n*** (where ***n*** is the number of processors to use)
    - For systemes with OpenMP installed, this is how the number of threads is specified

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_LITE or RSL (interface to MPI)
    - Use RSL_LITE if possible
      - Some newer features do not work with RSL
      - RSL going away in V3.0
  - 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.

Configuration File

- The ***configure.wrf*** file is built from three pieces within the ***arch*** directory

1. **preamble**: uniform requirement for the code, such as maximum number of domains, word size, etc.

2. **configure.defaults**: selection of compiler, parallel, communication layer
   - User edits if a change to the compilation options or library locations is needed

3. **postamble**: standard make rules and dependencies

Sample *configure.wrf*

```
FC              =       mpif90 -f90=pgf90
LD              =       mpif90 -f90=pgf90
CC              =       mpicc -cc=gcc -DMPI2_SUPPORT -DFSEEKO64_OK
SCC             =       gcc
SFC             =       pgf90
RWORDSIZE       =       $(NATIVE_RWORDSIZE)
CFLAGS          =       -DDM_PARALLEL
          -DMAXDOM_MAKE=$(MAX_DOMAINS)
          -DMAXPROC_MAKE=$(MAX_PROC) -I../external/RSL_LITE
FCOPTIM                 =       -O2 # -fast
FCDEBUG                 =       #-g
FCBASEOPTS      =       -w -byteswapio -Mfree $(FCDEBUG) # -Mlfs
FCFLAGS         =       $(FCOPTIM) $(FCBASEOPTS)
```

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
compile em_quarter_ss } ***3D Ideal Case (ARW only)***

compile em_grav2d_x
compile em_hill2d_x
compile em_squall2d_x
compile em_squall2d_y } ***2D Ideal Case (ARW only)***

compile em_real
compile nmm_real } ***Real Data Cases (ARW and NMM)***

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
    - ✓***nup.exe*** (not used much)
  - 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.

Clean Compilation

- To remove all object files (except those in *external/*) and executables, type:

  clean

- To remove all built files, including ***configure.wrf***, type:

  clean –a

  - Recommended if
    - compilation failed
    - registry changed
    - configuration file changed

Compiling both WRF cores

Using two different WRFV2 directory trees

Set environment variables for each and configure and compile as usual

Using the same WRFV2 directory tree

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 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*
- For simplicity, install WPS/ in the same location as WRFV2/
- After ***gunzip*** and ***untar***, should see a directory WPS/
  tar –zxvf WPSV2.2.1.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; GRIB2 only if needed)

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