WRF and WPS: Compile

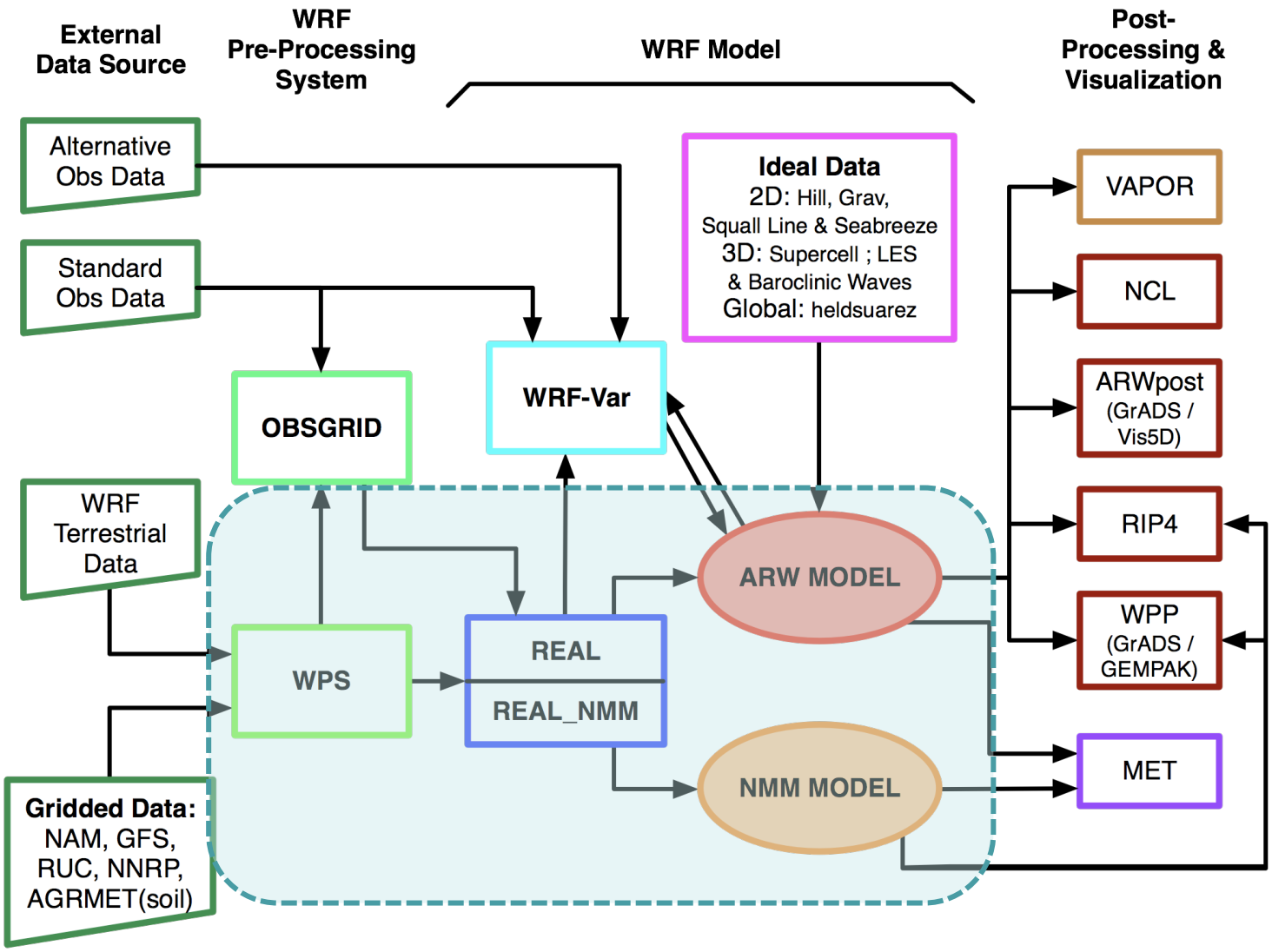
Laurie Carson
National Center for Atmospheric Research (NCAR)
The Developmental Testbed Center (DTC)

23 January, 2012

DTC Developmental Testbed Center

NSF

Program Flow



*First need to compile WRF and then WPS

System Requirements

Required libraries (WRF and WPS):

- FORTRAN 90/95 compiler
- C compiler
- Perl
- netCDF
- NCAR Graphics (optional, but recommended – used by graphical utility programs)

Optional libraries* for GRIB2 support (in WPS):

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

*Installation of these libraries is **NOT** part of the WPS installation script

Installing WRF

1. Download source code
2. Set environment
3. Configure and Compile WRF

1. Download WRF Source Code

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

  WRFV3.3.1.TAR.gz *(or the latest release available)*
- After ***gunzip*** and ***untar:***

  tar –zxvf WRFV3.3.1.TAR.gz
- look for a directory: **WRFV3/**

  cd to **WRFV3/** directory

WRFV3 Directory

Makefile	**Top-level makefile**
README	**General information about WRF code**
README.NMM	**NMM specific information**
README.rsl_output	**Describes options for RSL output files**
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**
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 →

2. 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 OMP_NUM_THREADS n*** (where ***n*** is the number of processors to use)
    - For systems with OpenMP installed, this is how the number of threads is specified
  - ***setenv MPICH_F90 f90*** (Or whatever FORTRAN compiler may be called)

3. Configuring WRF

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

  ./configure

- This script checks the system hardware and software, and then offers the user choices for configuring WRF:
  - Type of compiler
  - Serial, OpenMP, or MPI
  - Type of nesting (basic, preset moves, vortex following)

SAMPLE: List of Configure Options

Choices for IBM machines look like:

Please select from among the following supported platforms.

1. AIX xlf compiler with xlc (serial)
2. AIX xlf compiler with xlc (smpar)
3. AIX xlf compiler with xlc (dmpar)
4. AIX xlf compiler with xlc (dm+sm)

SAMPLE: List of Configure Options

Choices for 64-bit LINUX machines might look like:

Please select from among the following supported platforms.

1. Linux x86_64, PGI compiler with gcc (serial)
2. Linux x86_64, PGI compiler with gcc (smpar)
3. Linux x86_64, PGI compiler with gcc (dmpar)
4. Linux x86_64, PGI compiler with gcc (dm+sm)
5. Linux x86_64, PGI accelerator compiler with gcc (serial)
6. Linux x86_64, PGI accelerator compiler with gcc (smpar)
7. Linux x86_64, PGI accelerator compiler with gcc (dmpar)
8. Linux x86_64, PGI accelerator compiler with gcc (dm+sm)
9. Linux x86_64 i486 i586 i686, ifort compiler with icc (serial)
10. Linux x86_64 i486 i586 i686, ifort compiler with icc (smpar)
11. Linux x86_64 i486 i586 i686, ifort compiler with icc (dmpar)
12. Linux x86_64 i486 i586 i686, ifort compiler with icc (dm+sm)
13. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (serial)
14. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (dmpar)
15. x86_64 Linux, gfortran compiler with gcc (serial)
16. x86_64 Linux, gfortran compiler with gcc (smpar)
17. x86_64 Linux, gfortran compiler with gcc (dmpar)
18. x86_64 Linux, gfortran compiler with gcc (dm+sm)1dmpar)

List of Configure Options – Part 2

Compile for nesting? (0=no nesting, 1=basic, 2=preset moves, 3=vortex following) [default 0]:

Choices for Nesting are:

0. no nesting (only available for serial and smpar)
1. basic
2. preset moves
3. vortex following

- default is option 0 for serial/smpar, 1 for dmpar
- in addition, if running NMM with nesting:

setenv WRF_NMM_NEST 1

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, to test or adjust settings.
  - This file is overwritten each time "configure" is run. Any edits will be lost, unless they are made in the *arch/** files

Configuration File

- The ***configure.wrf*** file is built from three pieces within the ***arch*** directory
  1. **preamble_new**: uniform requirement for the code, such as maximum number of domains, word size, etc.
  2. **configure_new.defaults**: selection of compiler, parallel, communication layer
     - User edits if a change to the compilation options or library locations is needed
  3. **postamble_new**: standard make rules and dependencies

- The ***arch/configure_new.defaults*** file can be edited to add a new option if needed.

Sample *configure.wrf*

```
# Settings for Linux i486 i586 i686, PGI compiler with gcc  (dmpar)

DMPARALLEL  =       1
SFC         =       pgf90
SCC         =       gcc
DM_FC       =       mpif90 -f90=$(SFC)
DM_CC       =       mpicc -cc=$(SCC) -DMPI2_SUPPORT
FC          =       $(DM_FC)
CC          =       $(DM_CC) -DFSEEKO64_OK
LD          =      $(FC)
RWORDSIZE   =       $(NATIVE_RWORDSIZE)
FCOPTIM     =       -O2 -fast
FCNOOPT     =       -O0
FCDEBUG     =       # -g $(FCNOOPT)
```

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_heldsuarez	
compile em_les	
compile em_scm_xy	***2D Ideal Case (ARW only)***
compile em_grav2d_x	
compile em_hill2d_x	
compile em_squall2d_x	
compile em_squall2d_y	
compile em_seabreeze2d_x	
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 under **main/**:
  - ✓ ***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
    - want to compile different dynamic core
    - want to change configuration file (i.e. select a different compiler, options, etc)

Compiling both WRF cores

Using two different WRFV3 directory trees

Set environment variables for each and configure and compile as usual

Using the same WRFV3 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 WRFV3/

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 ***WRFV3/main*** (and to ***test/test_case/*** if you run there)

Installing WPS

1. Download static terrestrial data
2. Download source code
3. Set environment
4. 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.)
  - *modis_landuse_20class_30s* - MODIS landuse (Noah LSM only)
  - *orogwd* – data for gravity wave drag schemes
  - *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

Download WPS Source Code

- The WPS source code can be obtained from:
  http://www.mmm.ucar.edu/wrf/users/download/get_source.html

- WPS is designed to work with WRF
  - *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 WRFV3/

- After ***gunzip*** and ***untar***, should see a directory WPS/
  tar –zxvf WPSV3.3.1.TAR.gz *(or the latest release available)*

  ls
  WPS/ WRFV3/

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

Configure WPS

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

- This script offers the user choices for configuring WPS:
  - Type of compiler
  - Serial or Distributed memory
  - GRIB1 or GRIB2

- To use GRIB2 data, additional libraries are needed:
  setenv *JASPERINC /usr/local/jasper/include*
  setenv *JASPERLIB /usr/local/jasper/lib*

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

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

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***
  - ✓ ***calc_ecmwf_p.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
  - Make 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://forum.wrfforum.com*
  - WRF Email list: wrf_users@ucar.edu
  - WRF Help email: *wrfhelp@ucar.edu*