WRF and WPS: Compile

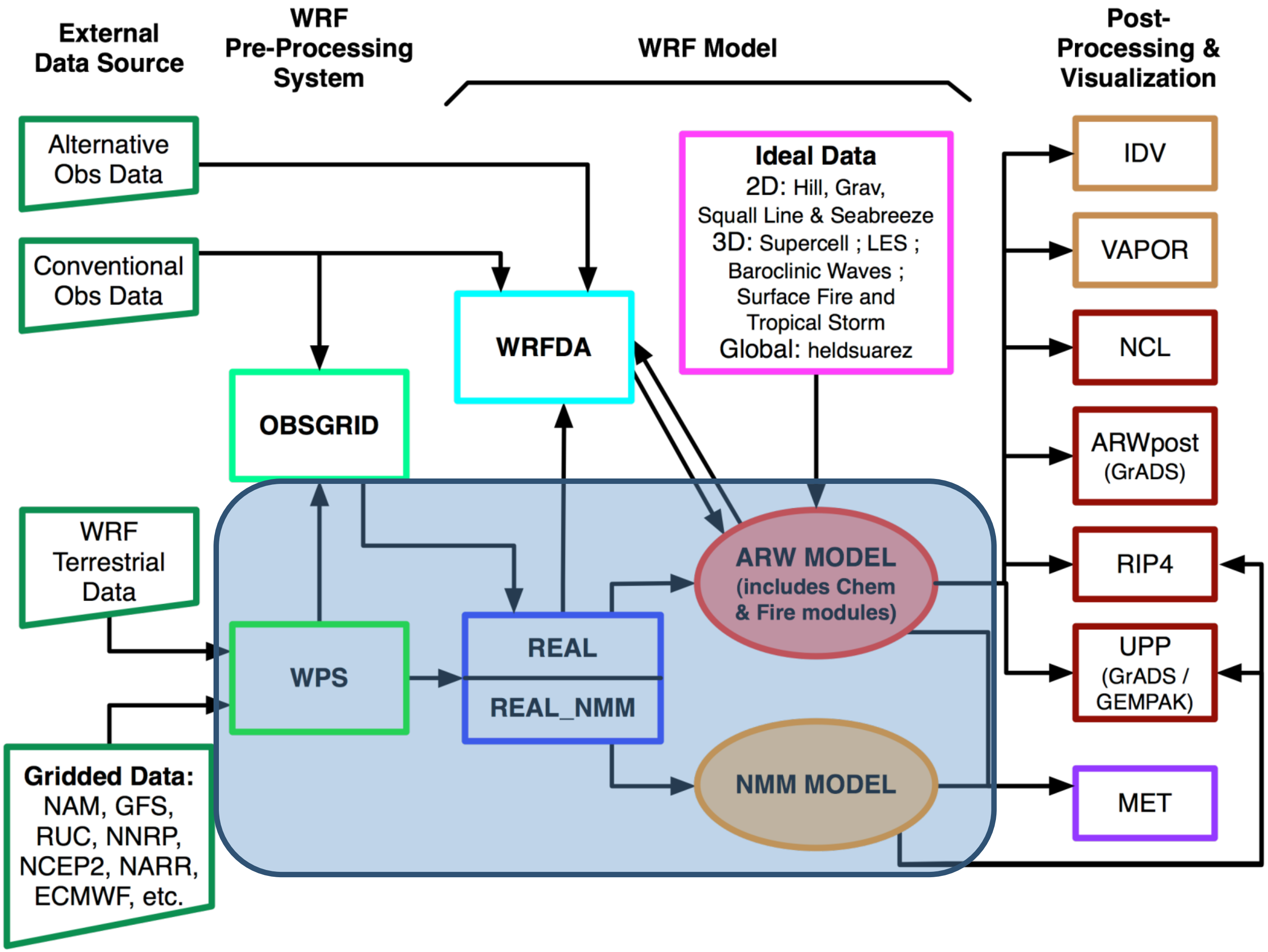
Laurie Carson

National Center for Atmospheric Research (NCAR)

The Developmental Testbed Center (DTC)

Developmental Testbed Center

WRF Modeling System Flow Chart


External Data Source
WRF Pre-Processing System
WRF Model
Post-Processing & Visualization
Alternative Obs Data
Conventional Obs Data
WRF Terrestrial Data
Gridded Data: NAM, GFS, RUC, NNRP, NCEP2, NARR, ECMWF, etc.
OBSGRID
WPS
WRFDA
REAL
REAL_NMM
Ideal Data
2D: Hill, Grav, Squall Line & Seabreeze
3D: Supercell ; LES ; Baroclinic Waves ; Surface Fire and Tropical Storm
Global: heldsuarez
ARW MODEL (includes Chem & Fire modules)
NMM MODEL
IDV
VAPOR
NCL
ARWpost (GrADS)
RIP4
UPP (GrADS / GEMPAK)
MET

Installation Steps

- Check System Requirements
- Download source codes
- Download static dataset
- Compile WRF first
- Compile WPS

System Requirements

- Required Compilers/languages:
  - FORTRAN 90/95 compiler
  - C compiler
  - Perl
- Required libraries:
  - 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
- Optional (recommended) MPI library:
  - e.g. mpich, mpich2, openmpi

System Requirements

- Installation of these libraries is not part of the WRF and WPS installation scripts
- Make sure these libraries are installed with the same FORTRAN compiler that is available to you to compile the WRF and WPS source code

Download 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
- Get the latest released code:

  WRFV3.TAR.gz

  WPSV3.TAR.gz
- Both the ARW and NMM cores are included

Additional Downloads

- Test Datasets
  - Output from WPS & WRF: useful for testing
  - Sample GRIB data for WPS (GFS)
- Terrain and Land Use datasets for WPS
  - Full resolution (30", 2', 5', 10')
  - Low resolution (10' only)
- download from the same site as the code:

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

Download Terrestrial Data

- The terrestrial fields interpolated by ***geogrid*** may be downloaded from same page as the code
- Data are static: only need to be downloaded once
- Data can be shared by users on the same machine by placing files in a common directory
- Data is not required for compilation, but is required at run-time
- The *geog.tar.gz* file contains the following data (~15GBuncompressed):

albedo_ncep	Monthly surface albedo
greenfrac	Monthly vegetation fraction
maxsnowalb	Maximum snow albedo
landuse USGS	24+1 categories
landuse MODIS	20+1 categories, NOAH LSM only
soiltemp	Annual mean deep soil temperature
soiltype_top	Top-layer soli type
soiltype_bot	Bottom-layer soil type
topo	Topography
orogwd	Subgrid orography informtion for gravity wave drag option
islope	Slope index (not used)
var_sso	Variance of subgrid-scale orography

Unzip and untar source code files

- Create a working directory and uncompress and untar both the WRF and WPS tarfiles

```
gunzip WRF.TAR.gz
tar –xf WRF.TAR.gz
gunzip WPS.TAR.gz
tar –xf WPS.TAR.gz
```

- After unzip and untar, you should have two directories:

```
WPS/
WRFV3/
```

WRFV3/ directory

Makefile	
README (multiple)	
clean compile configure	compile scripts
Registry/	data dictionary
arch/	compile rules
dyn_em/ dyn_exp/ dyn_nmm/ external/ frame/ inc/ main/ phys/ share/ tools/	source code directories
run/ test/	run directories

WPS/ directory

Makefile
README (multiple)

clean
compile
configure

compile scripts

arch/

compile rules

geogrid/
ungrib/
metgrid/

source code directories

util/

utilities

link_grib.csh
namelist.wps
namelist.wps_all-options

run time options

Set environment variables

- If the ***netCDF*** is not in the standard ***/usr/local*** then set the **NETCDF** environment variable

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

- WRF needs both the ***lib*** and ***include*** directories
- As a general rule, make sure the ***netCDF*** and ***MPI*** libraries are installed using the same compiler that will be used to compile WRF

e.g. PGI, Intel, gfortran

Compile WRF first…

There are two steps:

1) Create a configuration file for your computer and compiler

   ./configure

2) Compile the code

   ./compile test_case

Create the configuration file

./configure

This script checks the system hardware and software, and then offers the user choices for configuring WRF:

- Type of compiler
- Serial, OpenMP (smpar), MPI (dmpar) or MPI+OpenMP (dm+sm)
- Type of nesting (basic, preset moves, vortex following)

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)
```

Choices for Nesting are:

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

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
```

WRF configuration file

- 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.
  - If you wish to change compiler settings for one-time only, edit ***configure.wrf***
  - If you wish to change default compiler settings, edit ***arch/configure_new.defaults***

Sample configure.wrf

```
# Settings for Linux x86_64, PGI compiler with gcc  (dmpar)
#
DMPARALLEL      =       1
OMPCPP          =       # -D_OPENMP
OMP             =       # -mp -Minfo=mp -Mrecursive
OMPCC           =       # -mp
SFC             =       pgf90
SCC             =       gcc
CCOMP           =       pgcc
DM_FC           =       mpif90
DM_CC           =       mpicc -cc=$(SCC) -DMPI2_SUPPORT
FC              =       $(DM_FC)
CC              =       $(DM_CC) -DFSEEKO64_OK
LD              =       $(FC)
RWORDSIZE       =       $(NATIVE_RWORDSIZE)
PROMOTION       =       -r$(RWORDSIZE) -i4
ARCH_LOCAL      =       -DNONSTANDARD_SYSTEM_SUBR
CFLAGS_LOCAL    =       -w -O3
LDFLAGS_LOCAL   =
CPLUSPLUSLIB    =
ESMF_LDFLAG     =       $(CPLUSPLUSLIB)
FCOPTIM         =       -O3 #-fastsse -Mvect=noaltcode -Msmartalloc -Mprefetch=distance:8 -Mfprelaxed
FCREDUCEDOPT    =       $(FCOPTIM)
FCNOOPT         =       -O0
FCDEBUG         =       # -g $(FCNOOPT)
```

Compile 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

Compile 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
compile em_heldsuarez
compile em_les
compile em_scm_xy
compile em_grav2d_x
compile em_hill2d_x
compile em_squall2d_x
compile em_squall2d_y
compile em_seabreeze2d_x
compile em_real
compile nmm_real

compile -h *help message*

More on Compile…

- Compiling WRF will take 20-30 minutes
- Since V3.2, parallel make is supported if "make" on your computer supports it
- Two processors are used by default. If you would like to change it, set the environment variable before compile:

```
setenv J "-j 1"
```

WRF executables

If compile is successful, you should find these executables in WRFV3/main

- em_real:
  - wrf.exe – model executable
  - real.exe – ARW initialization
  - ndown.exe – one-way nesting
  - tc.exe – for TC bogusing (serial only)
- nmm_real:
  - wrf.exe – model executable
  - real_nmm.exe – NMM initialization
- Any idealized case:
  - wrf.exe – model executable
  - ideal.exe – ideal case initialization
  - Each ideal test case creates different executables

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

- Use two different WRFV3 directory trees
  - Set environment variables for each and configure and compile as usual

- Using the same WRFV3 directory tree
  - Core "A" – configure and compile
    - Save ***main/wrf.exe*** to ***main/wrf_coreA.exe***
    - Copy ***main/*exe*** to a temporary location outside of WRFV3/
  - ***clean –a***
  - Core "B" – configure and compile
    - Save ***wrf.exe*** to ***wrf_coreB.exe***

Compile WPS…

There are two steps:

1) Create a configuration file for your computer and compiler

   ./configure

2) Compile the code

   ./compile

Create the configuration file

In the WPS/ directory:

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

- This script offers the user choices for configuring WPS:
  - Type of compiler
  - Serial or MPI (dmpar)
  - with or without GRIB2 support

- To use GRIB2 data, additional libraries are needed:

  setenv *JASPERINC /usr/local/jasper/include*
  setenv *JASPERLIB /usr/local/jasper/lib*

Choices for 64-bit LINUX machines might look like:

```
Please select from among the following supported platforms.

  1.  Linux x86_64, gfortran    (serial)
  2.  Linux x86_64, gfortran    (serial_NO_GRIB2)
  3.  Linux x86_64, gfortran    (dmpar)
  4.  Linux x86_64, gfortran    (dmpar_NO_GRIB2)
  5.  Linux x86_64, PGI compiler   (serial)
  6.  Linux x86_64, PGI compiler   (serial_NO_GRIB2)
  7.  Linux x86_64, PGI compiler   (dmpar)
  8.  Linux x86_64, PGI compiler   (dmpar_NO_GRIB2)
  9.  Linux x86_64, PGI compiler, SGI MPT   (serial)
 10.  Linux x86_64, PGI compiler, SGI MPT   (serial_NO_GRIB2)
 11.  Linux x86_64, PGI compiler, SGI MPT   (dmpar)
 12.  Linux x86_64, PGI compiler, SGI MPT   (dmpar_NO_GRIB2)
 13.  Linux x86_64, IA64 and Opteron    (serial)
 14.  Linux x86_64, IA64 and Opteron    (serial_NO_GRIB2)
 15.  Linux x86_64, IA64 and Opteron    (dmpar)
 16.  Linux x86_64, IA64 and Opteron    (dmpar_NO_GRIB2)
 17.  Linux x86_64, Intel compiler    (serial)
 18.  Linux x86_64, Intel compiler    (serial_NO_GRIB2)
 19.  Linux x86_64, Intel compiler    (dmpar)
 20.  Linux x86_64, Intel compiler    (dmpar_NO_GRIB2)
 21.  Linux x86_64 g95 compiler     (serial)
 22.  Linux x86_64 g95 compiler     (serial_NO_GRIB2)
 23.  Linux x86_64 g95 compiler     (dmpar)
 24.  Linux x86_64 g95 compiler     (dmpar_NO_GRIB2)
```

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*)

WPS utilities

- 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 a WPS installation

- A single build of WPS will work for both the ARW and NMM cores
- Multiple users may share a single installation of the WPS; not every user needs to install it
  - 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*