SIMPLIFYING END-TO-END NUMERICAL MODELING USING SOFTWARE CONTAINERS

Users' Page:

https://dtcenter.org/community-code/numerical-weather-prediction-nwp-containers

Online Tutorial:

https://dtcenter.org/container_nwp_tutorial/index.php

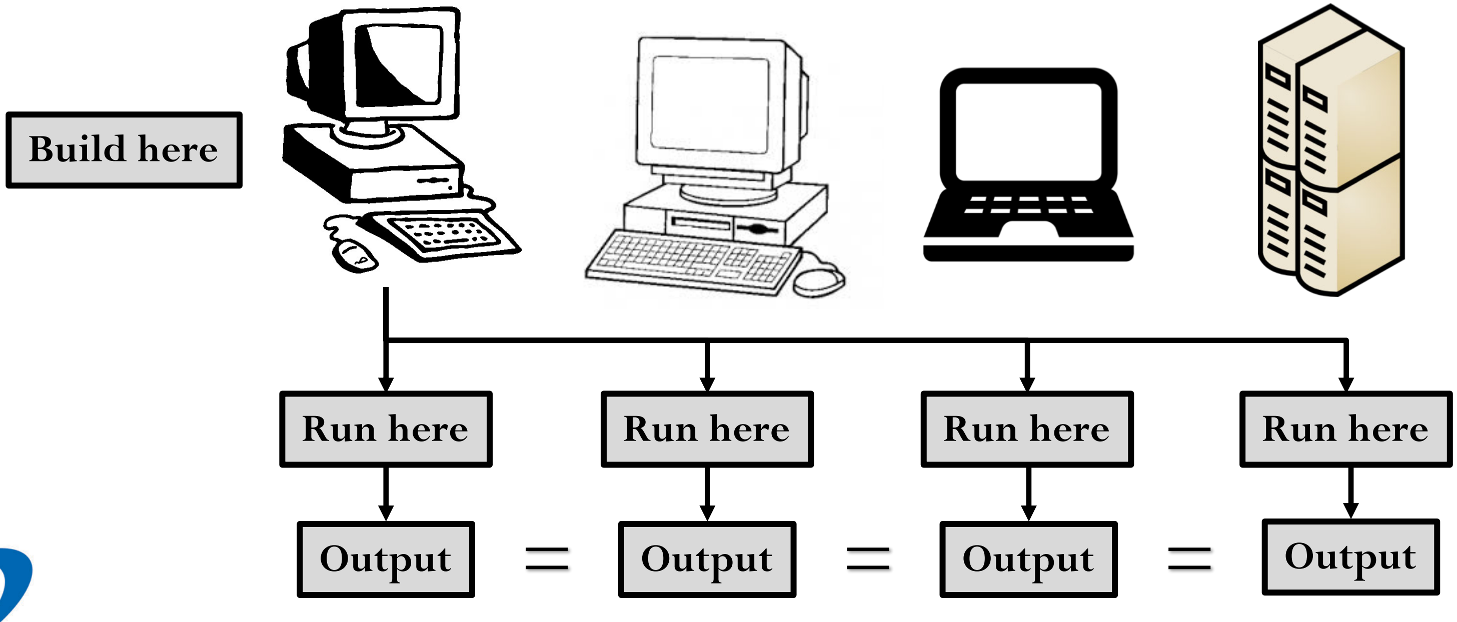
Developmental Testbed Center

Why use containers?

Introduction to Docker

What is a container?

- A container is a self-contained “box” that allows you to essentially build software once and run it anywhere
- Similar to a virtual machine but much more lightweight and portable



Build here

Run here

Run here

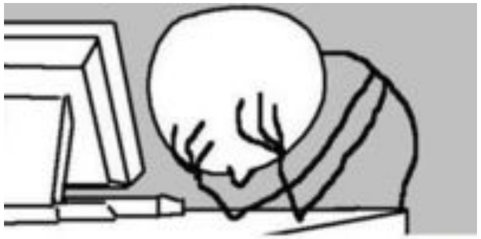
Run here

Run here

Output = Output = Output = Output

Why use containers?

- Software systems require substantial set-up to get all the necessary code, including external libraries, compiled on a specific platform → ***Containers help solve this problem!***
  - Efficient, lightweight, secure, and self-contained (including operating system, libraries, code, and executables) systems
  - Everything required is packaged into isolated components, ready for development, shipment, and deployment directly to users
  - Software should always run the same, regardless of where it is deployed
  - Eliminates possible frustrations with up-front system setup
  - Facilitates cloud computing, classroom use, etc.



Someone trying to compile WRF

What is a Docker container?

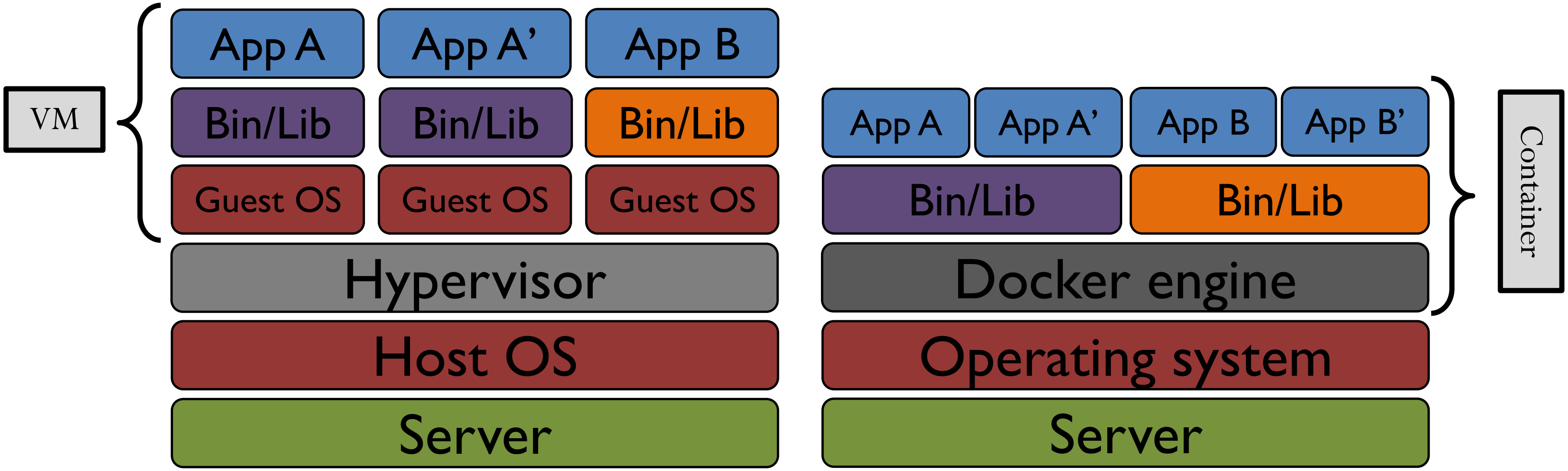
- Docker is one of the leading software containerization platforms
    - Home page: **https://www.docker.com**
    - Documentation: **https://docs.docker.com**
- A Docker container
    + is open source
    + is an easy way to build a development environment
    + can hold applications “inside the container”
    + is portable across Linux, Mac, and Windows machines
    + is much smaller than a virtual machine
    + sets up a partition between the host machine and “container land”
    + allows “root” inside the container but doesn’t alter permissions on the host machine
    - ***requires root access to install and update Docker***
    - ***running on more than a single node is cumbersome***

Docker disadvantages, Singularity solutions?

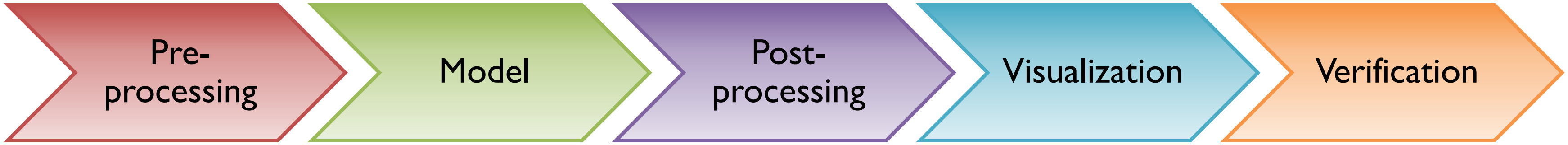
- A newer container software Singularity sets out to eliminate these issues:
  - Designed for HPC applications
  - Has easy support for cross-node MPI applications
  - Docker containers can be converted for use with Singularity
- Charlieloud and Shifter
  - Designed for HPC applications
  - Based on Docker, but less user-friendly

Virtual machine vs. containers

- Containers vs. virtual machines: VMs bundle a full operating system, whereas containers only contain necessary libraries and dependencies



Why use containers for NWP?



Advantages:

- ✓ Reduces spin-up time to build necessary code components
- ✓ Highly portable
- ✓ Easily sharable with other collaborators
- ✓ Use in cloud computing
- ✓ Easy to replicate procedures and results

Who can benefit from containers?

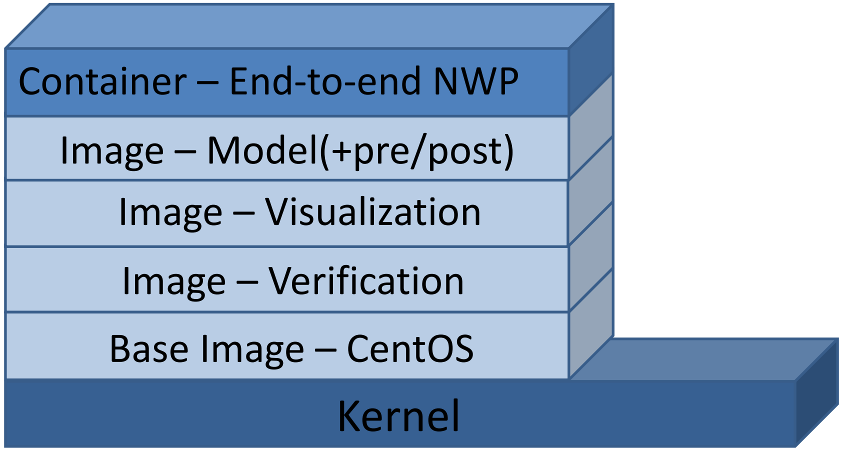
- ✓ Graduate and undergrad students
- ✓ University faculty
- ✓ Researchers
- ✓ Tutorial participants
- ✓ Use for community support

Ultimately, containers will substantially reduce the spin-up time with setting up software systems, which promotes greater efficiency in producing model and statistical output!

Understanding the lingo: Images & containers

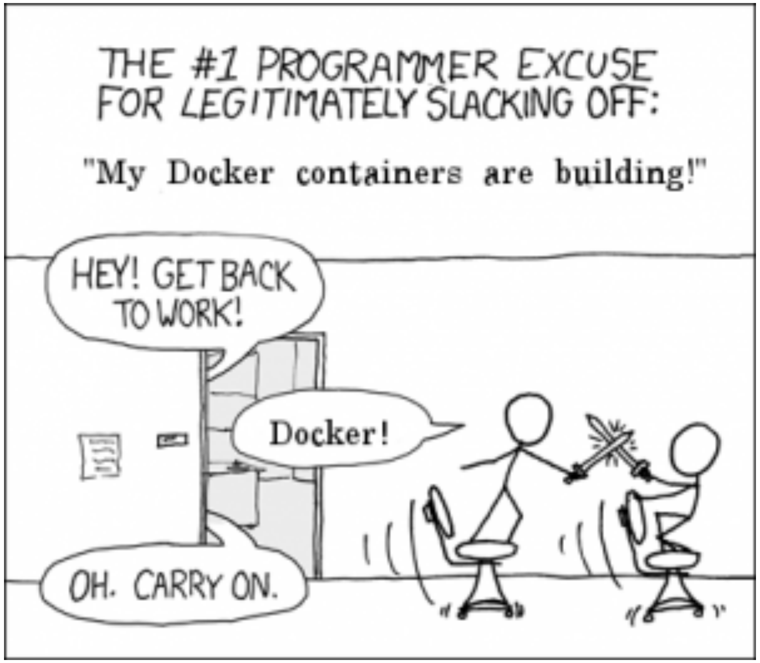
- Image:
  - Inert, immutable snapshot
  - Created with the build command
  - Will produce a container when started with run command
- Container:
  - Instance of an image
  - Created with the run command
  - Can have many running containers of the same image

Example of containerized end-to-end system



Building, saving, and loading images

- Images can be…
  - Built from scratch *(slower, but offers customization!)*
  - Saved to a tar file, which can then be loaded for faster deployment


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"My Docker containers are building!"
HEY! GET BACK TO WORK!
Docker!
OH. CARRY ON.

Helpful Docker commands

- docker build -t my-name . : builds image
- docker images : see what images are built
- docker rmi : remove image
- docker save my-name > my-name.tar.gz : save an image
- docker load < my-name.tar.gz : load a saved image

- docker run --rm -it --volumes-from -v --name : run a command in a new container
  - --rm: Automatically remove the container when it exits
  - -it: create an interactive bash shell in the container
  - --volumes-from: Mount volumes from the specified container(s)
  - -v: Bind mount a volume
  - --name: Assign a name to the container
- docker ps -a : see what containers are running; obtain container ID
- docker rm : remove container using ID

HPC and cloud computing

- Models keep trending toward higher resolution and the use of ensembles is becoming more prevalent
  - Computational resources and increases in data volumes make it difficult to run containerized NWP systems with Docker on a single node → Running containers on HPC and cloud-computing platforms is becoming more appealing
- HPC and cloud computing offers excellent opportunity for providing a sandbox to the community to share output and datasets, allowing for cross-community interactions

DTC containers

Available containers

Case study overview

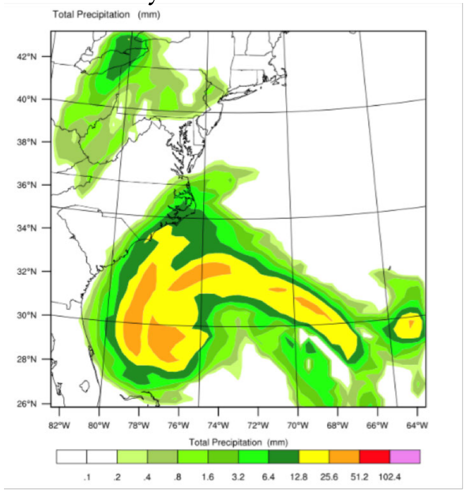
What is in the DTC containers?

- DTC containers package everything that is needed to build and run the model and produce graphics and verification, including code and data
  - Initial capability heavily leveraged from Big Weather Web (http://bigweatherweb.org/) initiative to containerize WRF (Hacker et al. 2017)
  - Uses gfortran; can be run serially or with distributed memory
- README files with explicit instructions for building and running WPS, WRF, GSI, UPP, NCL, MET, and METViewer
- Necessary namelist and configuration files
  - Vtable.GFS
  - namelist.wps and namelist.input
  - MET configuration files
- Case-specific data
  - GFS files for ICs/LBCs
  - Observation data for gridded (Stage II) and point (NDAS prepbufr) verification and data assimilation

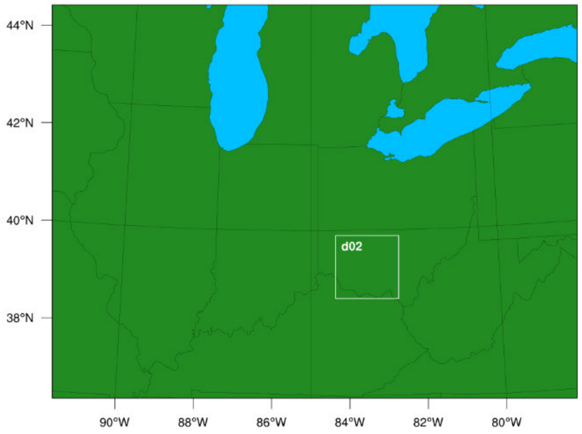
What cases are provided?

- Three cases with full datasets are provided in current inventory
  - Hurricane Sandy (Initialized on 27 Oct. 2012)
    - 40-km domain centered over East Coast (6-h forecast)
  - Derecho event over the Eastern CONUS (Initialized on 29 June 2012)
    - 12-km parent domain with 3-km nest over southern Ohio (24-h forecast)
  - Snow case across the mid-Atlantic region (Initialized on 23 Jan. 2016)
    - 30-km CONUS domain (24-h forecast)

Total precip for Hurricane Sandy model forecast

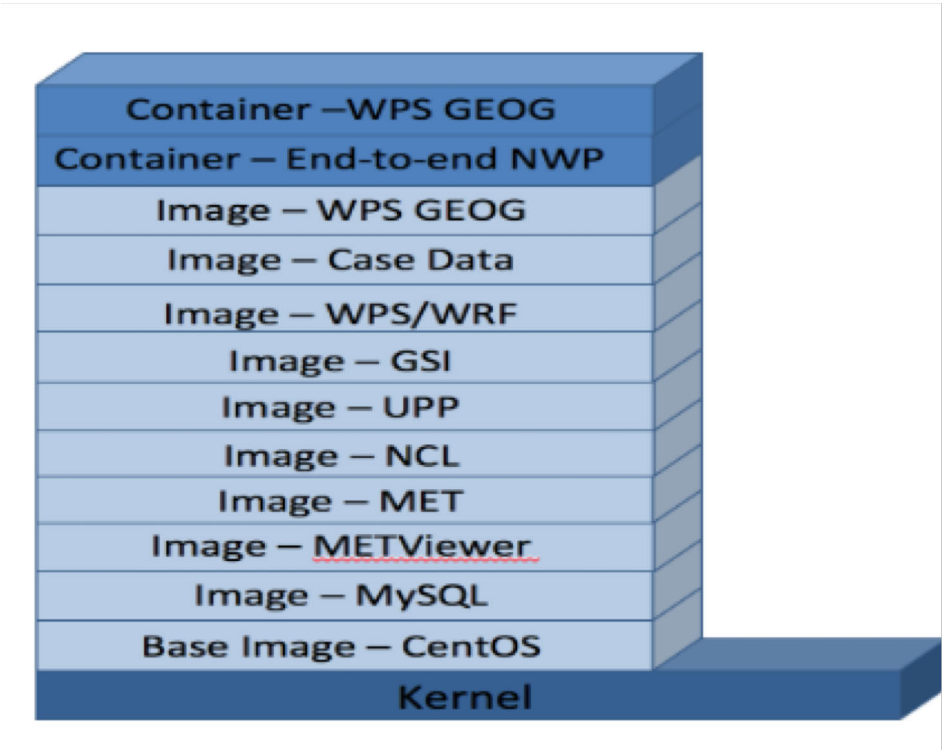


Derecho case domain configuration



What does this look like for the end-to-end DTC container?

Contains necessary build and run commands to run end-to-end NWP workflow



Container –WPS GEOG

Container – End-to-end NWP

Image – WPS GEOG

Image – Case Data

Image – WPS/WRF

Image – GSI

Image – UPP

Image – NCL

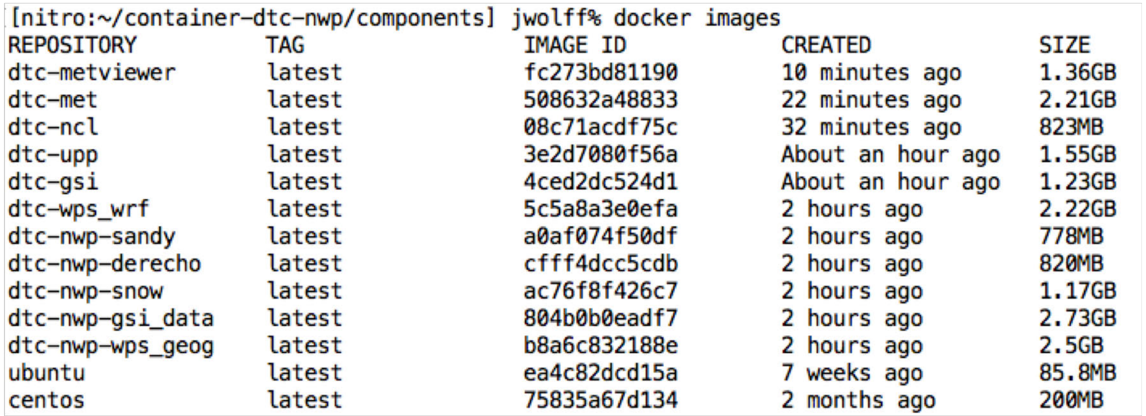
Image – MET

Image – METViewer

Image – MySQL

Base Image – CentOS

Kernel



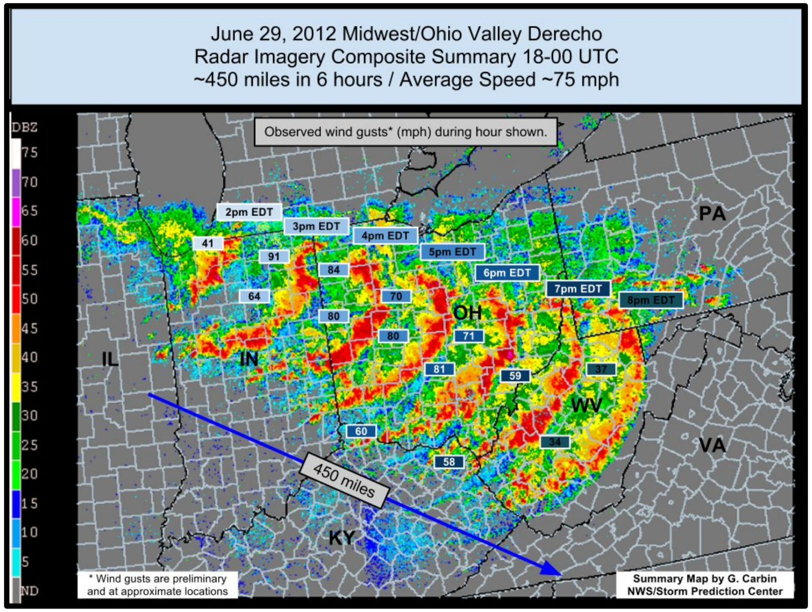
```
[nitro:~/container-dtc-nwp/components] jwolff% docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dtc-metviewer       latest              fc273bd81190        10 minutes ago      1.36GB
dtc-met             latest              508632a48833        22 minutes ago      2.21GB
dtc-ncl             latest              08c71acdf75c        32 minutes ago      823MB
dtc-upp             latest              3e2d7080f56a        About an hour ago   1.55GB
dtc-gsi             latest              4ced2dc524d1        About an hour ago   1.23GB
dtc-wps_wrf         latest              5c5a8a3e0efa        2 hours ago         2.22GB
dtc-nwp-sandy       latest              a0af074f50df        2 hours ago         778MB
dtc-nwp-derecho     latest              cfff4dcc5cdb        2 hours ago         820MB
dtc-nwp-snow        latest              ac76f8f426c7        2 hours ago         1.17GB
dtc-nwp-gsi_data    latest              804b0b0eadf7        2 hours ago         2.73GB
dtc-nwp-wps_geog    latest              b8a6c832188e        2 hours ago         2.5GB
ubuntu              latest              ea4c82dcd15a        7 weeks ago         85.8MB
centos              latest              75835a67d134        2 months ago        200MB
```

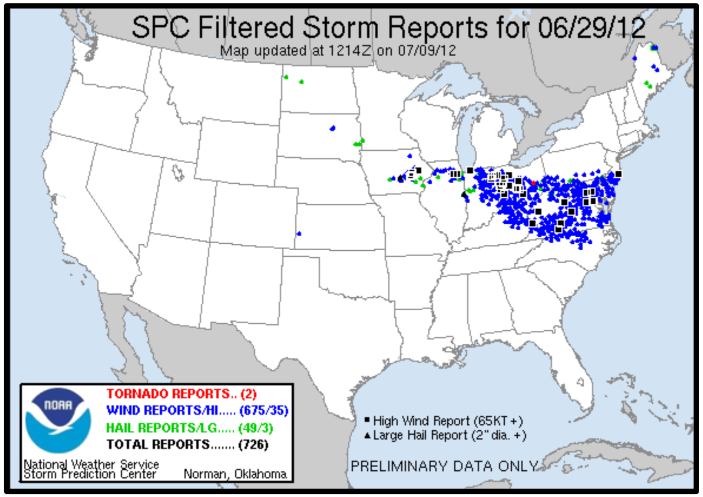
DTC Container Links

- **DTC end-to-end components**
  - **https://github.com/NCAR/container-dtc-nwp/**
- **MET**
  - **https://github.com/NCAR/container-dtc-met/**
  - **https://dtcenter.org/met/users/downloads/docker_container.php** (Instructions for installing and running pre-built container)
- **METViewer (Containers for MySQL and METViewer)**
  - **https://github.com/NCAR/container-dtc-metviewer/**
- **End-to-end NWP container online tutorial**
  - **https://dtcenter.org/met/docker-nwp/tutorial/container_nwp_tutorial/index.php**

Derecho Event Background

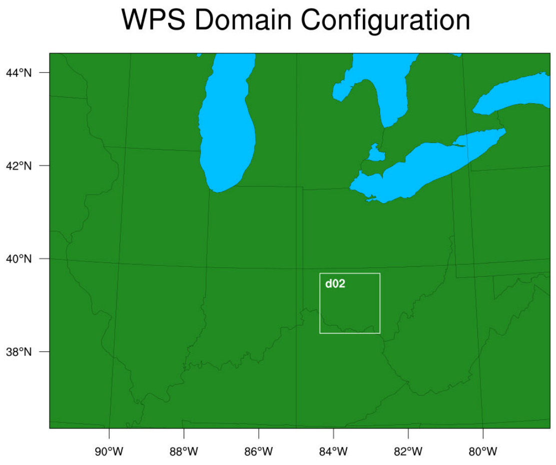
- Progressive derecho originated in Midwest, moved ESE across the Ohio Valley into the Mid-Atlantic
  - Traversed over 700 miles over 10 states
  - 13 deaths directly associated with storm
  - 4 million lost power
- Operational forecast guidance:
  - Global Forecast System (GFS) and NAM did not provide much forecast assistance more than 24 hours out from the event
  - High-Resolution Rapid Refresh (HRRR) model forecast an MCS to move through impacted area on morning of 29 June 2012 → *however, previous performance by HRRR did not allow for much confidence in forecast*

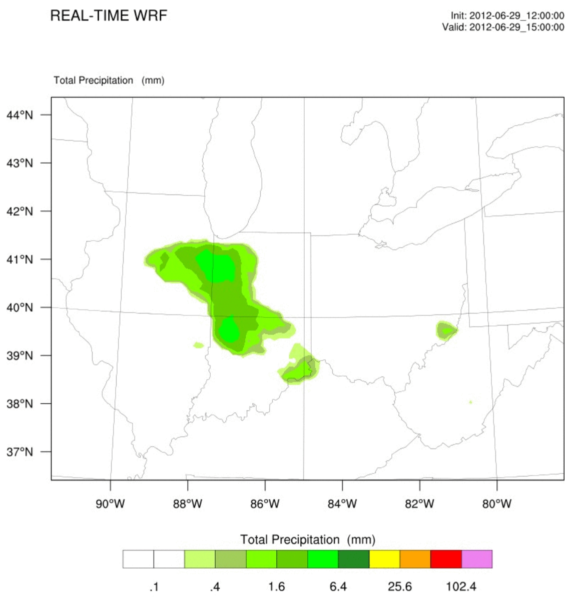

June 29, 2012 Midwest/Ohio Valley Derecho
Radar Imagery Composite Summary 18-00 UTC
~450 miles in 6 hours / Average Speed ~75 mph
DBZ
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
ND
Observed wind gusts* (mph) during hour shown.
2pm EDT
3pm EDT
4pm EDT
5pm EDT
6pm EDT
7pm EDT
8pm EDT
41
91
84
64
70
80
80
71
81
59
37
60
34
58
PA
OH
IL
IN
WV
VA
KY
450 miles
* Wind gusts are preliminary
and at approximate locations
Summary Map by G. Carbin
NWS/Storm Prediction Center


SPC Filtered Storm Reports for 06/29/12
Map updated at 1214Z on 07/09/12
NOAA
TORNADO REPORTS.. (2)
WIND REPORTS/HI..... (675/35)
HAIL REPORTS/LG..... (49/3)
TOTAL REPORTS...... (726)
National Weather Service
Storm Prediction Center
Norman, Oklahoma
■ High Wind Report (65KT+)
▲ Large Hail Report (2" dia. +)
PRELIMINARY DATA ONLY

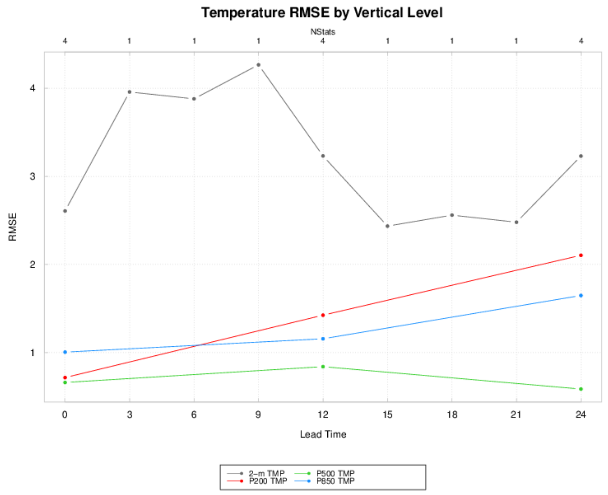
DTC Developmental Testbed Center

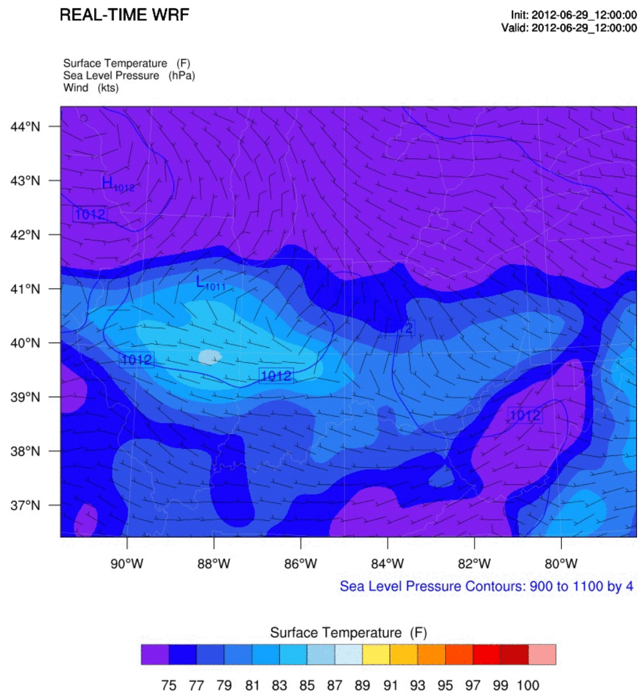
Derecho Event Output

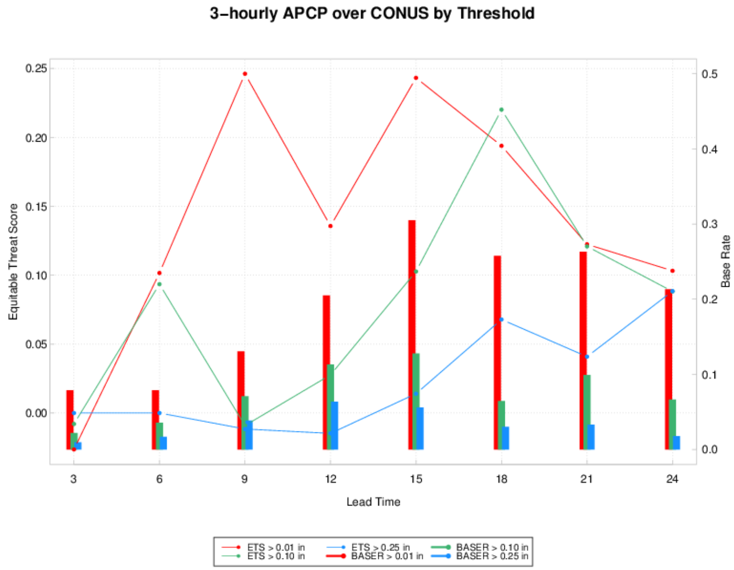




OUTPUT FROM WRF V3.9.1.1 MODEL
WE = 101 ; SN = 76 ; Levels = 51 ; Dis = 12km ; Phys Opt = 8 ; PBL Opt = 5 ; Cu Opt = 3

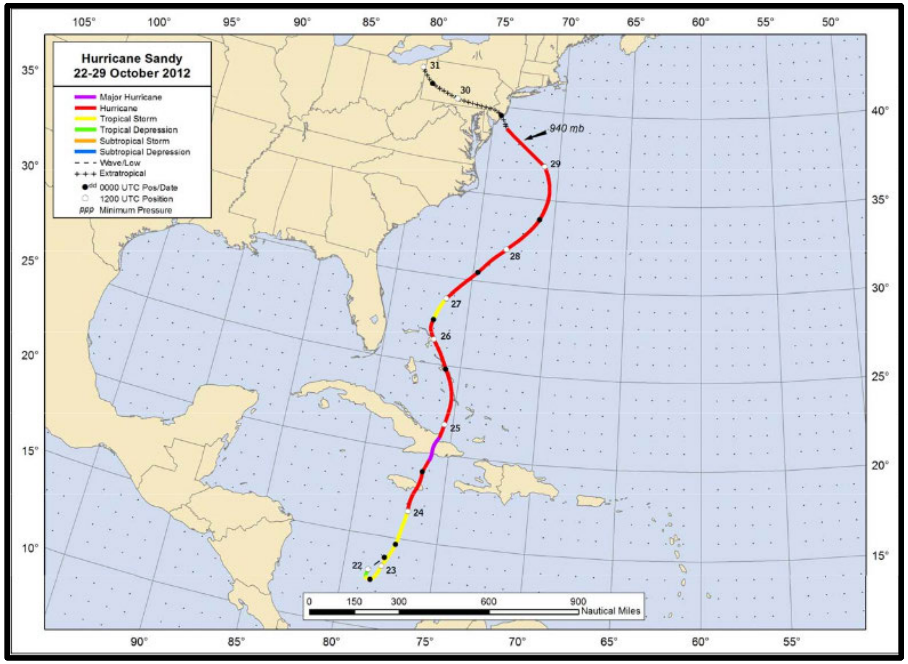


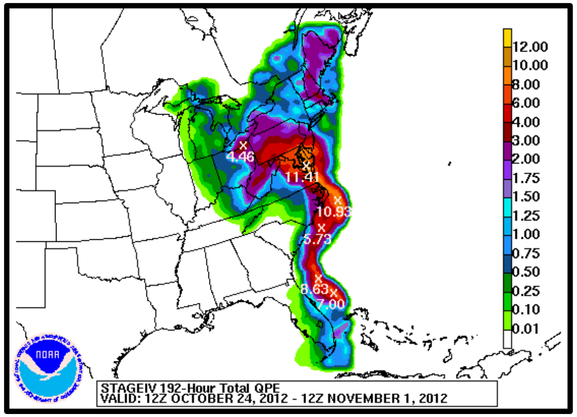




Sandy Event Background

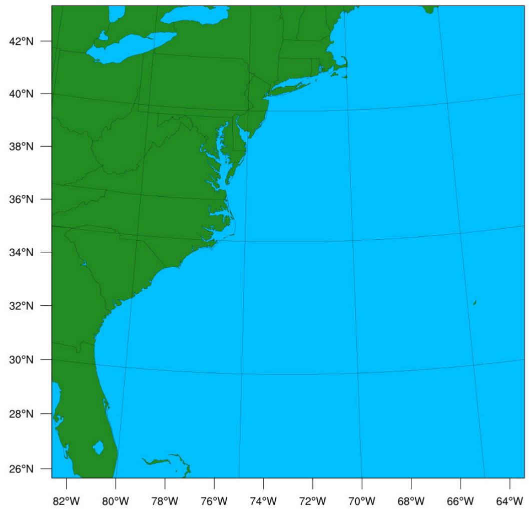
- Most deadly and destruction hurricane during the 2012 Atlantic hurricane season
  - 230 direct/indirect fatalities
  - Category 1 over Jamaica
  - Impacted Cuba as a Category 3
  - Weakened to a Category 1 over the Bahamas
  - Curved northwest and came onshore in New Jersey with hurricane force winds (no longer tropical)

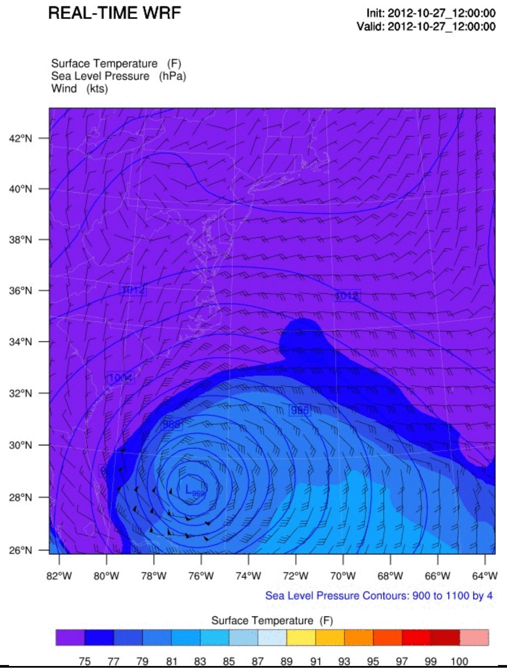


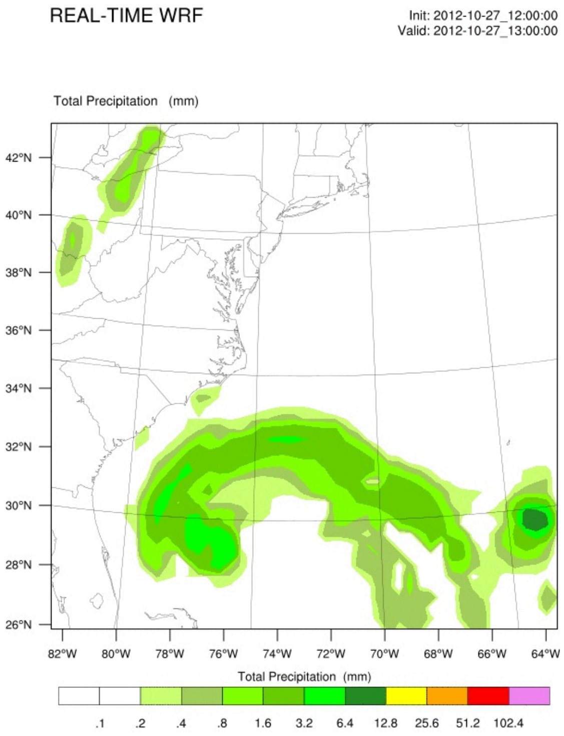


Sandy Event Output

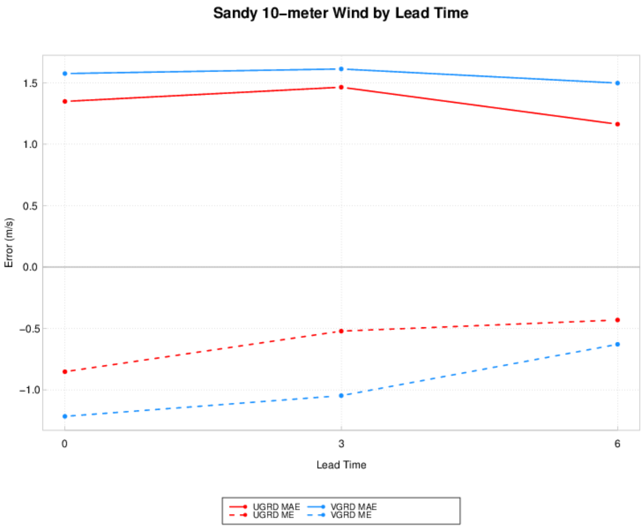
WPS Domain Configuration

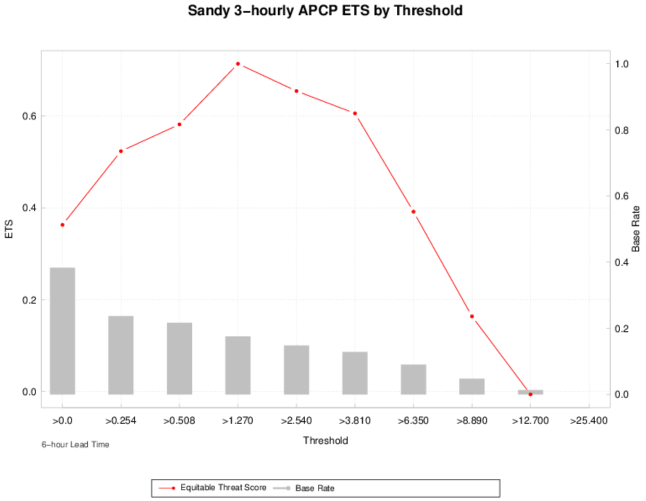






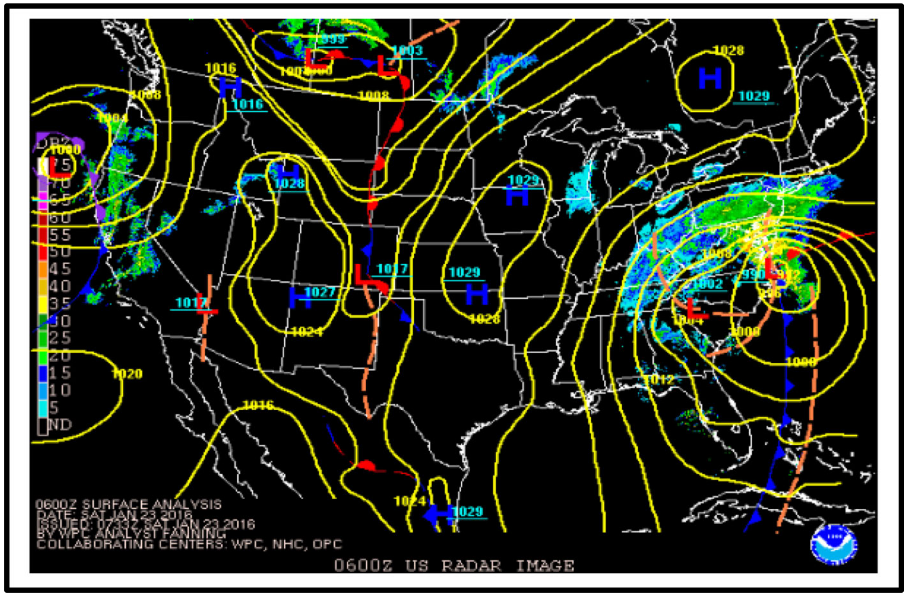
OUTPUT FROM WRF V3.9.1.1 MODEL
WE = 50 ; SN = 50 ; Levels = 60 ; Dis = 40km ; Phys Opt = 4 ; PBL Opt = 1 ; Cu Opt = 1

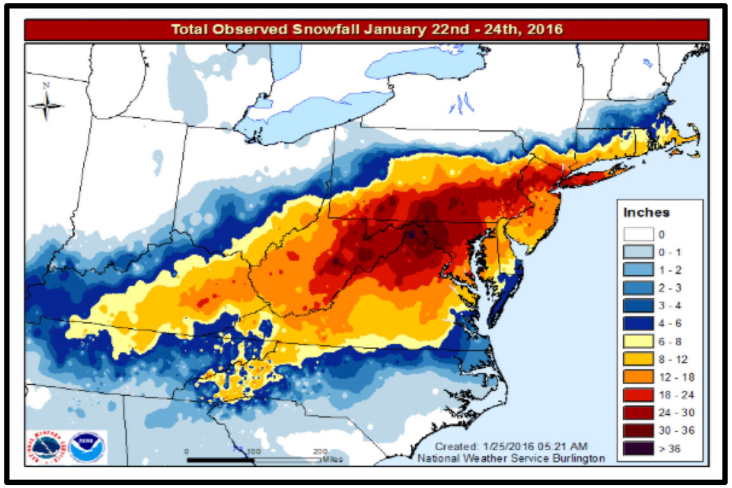




Snow Event Background

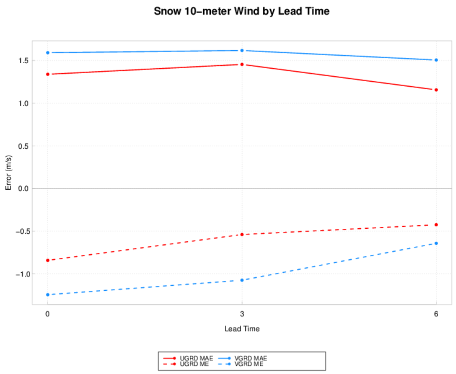
- Classic set-up for a major winter storm
  - Developed near the Gulf Coast and strengthened rapidly as it moved slowly up the coast
  - Produced significant amounts of snow, sleet, and freezing rain
  - Maximum amounts of 30-42" of snow in the mountains near the border of VA/WV/MD

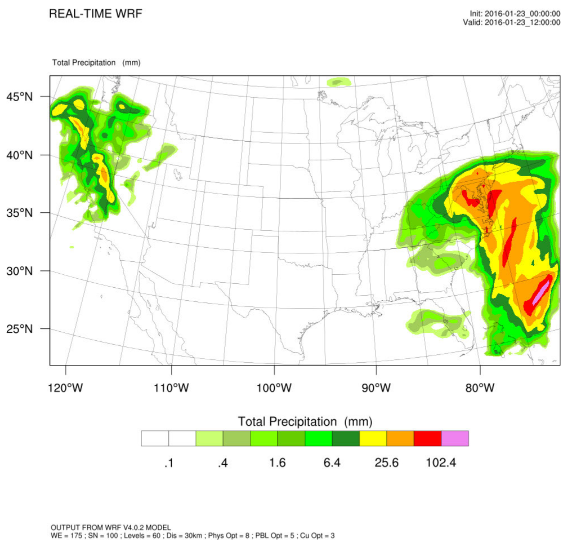




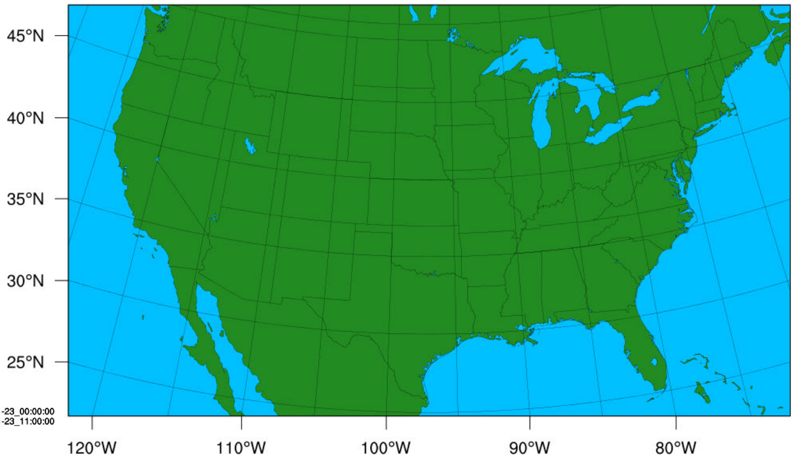
Total Observed Snowfall January 22nd - 24th, 2016
N
Inches
0
0 - 1
1 - 2
2 - 3
3 - 4
4 - 6
6 - 8
8 - 12
12 - 18
18 - 24
24 - 30
30 - 36
> 36
0
100
200
Miles
Created: 1/25/2016 05:21 AM
National Weather Service Burlington

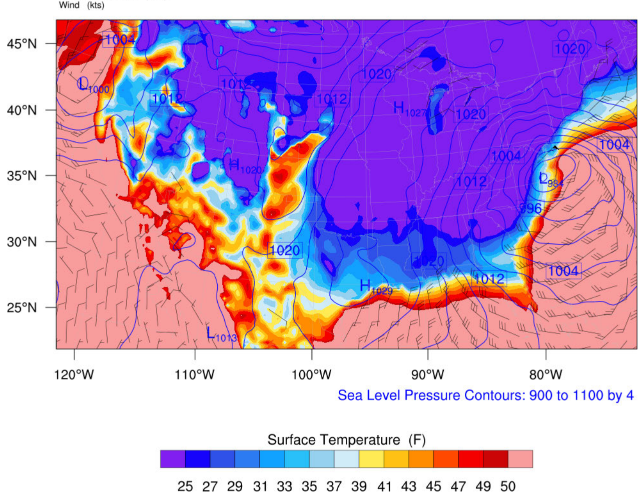
Snow Event Output

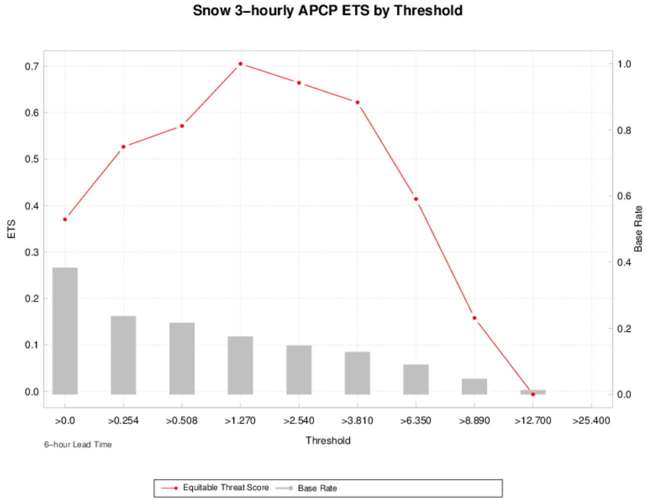



REAL-TIME WRF
Init: 2016-01-23_00:00:00
Valid: 2016-01-23_12:00:00
Total Precipitation (mm)
45°N
40°N
35°N
30°N
25°N
120°W
110°W
100°W
90°W
80°W
Total Precipitation (mm)
.1
.4
1.6
6.4
25.6
102.4
OUTPUT FROM WRF V4.0.2 MODEL
WE = 175 ; SN = 100 ; Levels = 60 ; Dis = 30km ; Phys Opt = 8 ; PBL Opt = 5 ; Cu Opt = 3

WPS Domain Configuration





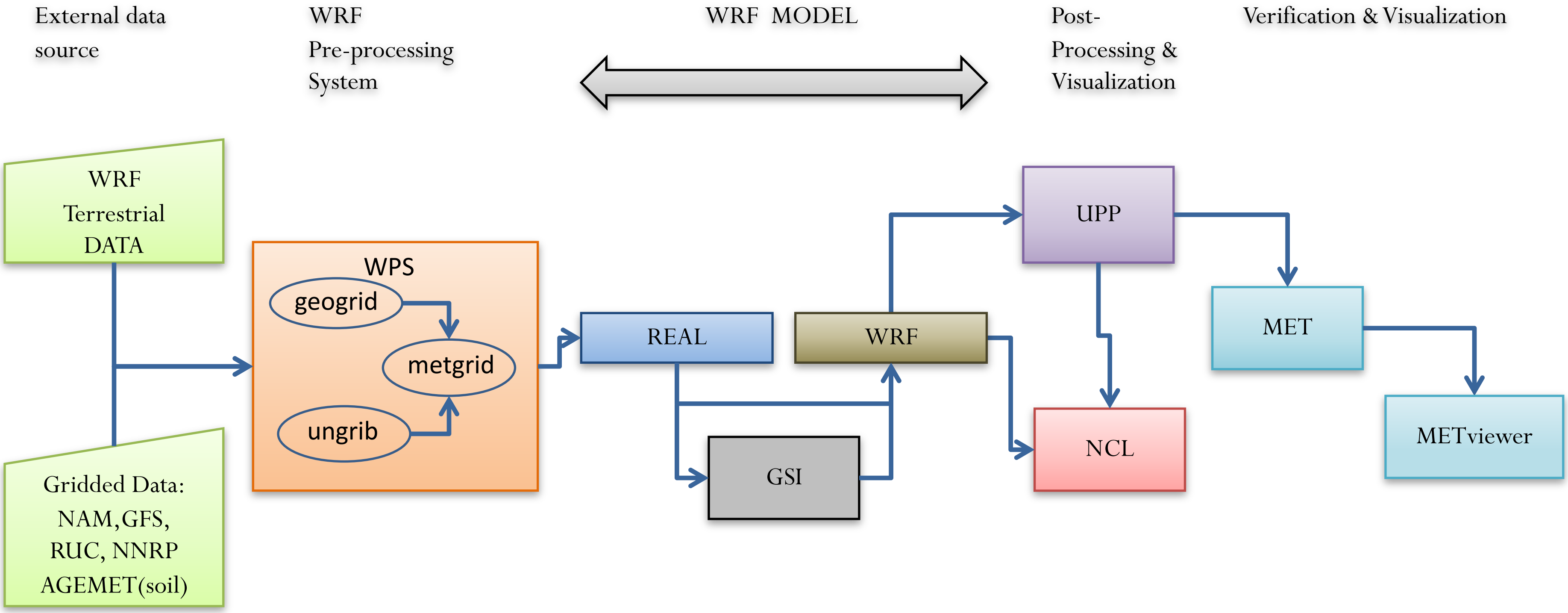


Introduction to containerized NWP, visualization, and verification

WPS, GSI, WRF, UPP, NCL, MET, METviewer

Containerized NWP, visualization, and verification

Introduction to WPS, GSI, WRF, UPP, NCL, MET, METviewer


External data source
WRF Pre-processing System
WRF MODEL
Post-Processing & Visualization
Verification & Visualization
WRF Terrestrial DATA
Gridded Data: NAM,GFS, RUC, NNRP AGEMET(soil)
WPS
geogrid
metgrid
ungrib
REAL
WRF
GSI
UPP
NCL
MET
METviewer

What is WPS?

- The WRF Preprocessing System (WPS) takes existing 4-d atmospheric data from GRIB-format files and interpolates it onto the user's specified WRF domain grid.
  - Initial conditions: 3-dimensional wind, temperature, geopotential height and RH, 2-dimensional surface pressure
  - Boundary conditions for the parent domain for the full length of the forecast
  - Commonly used input datasets are forecast output from GFS, or re-analysis datasets from NCEP or ECMWF

Global input data

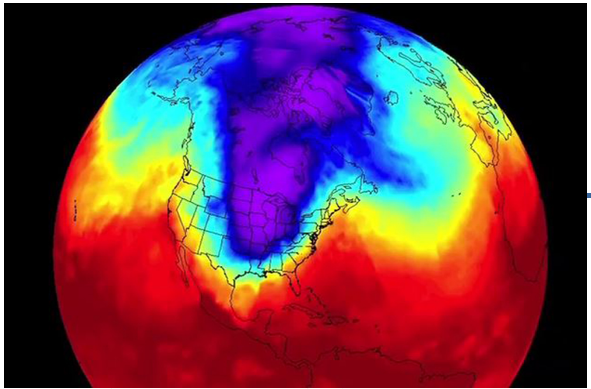
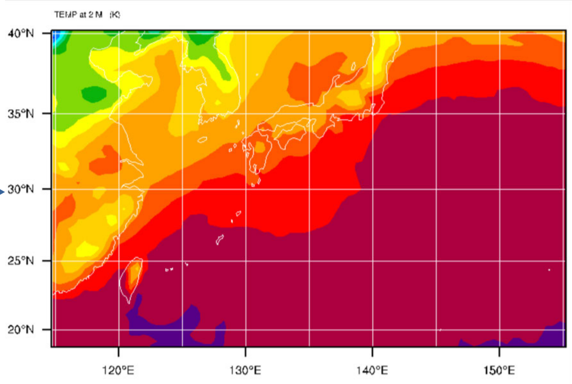


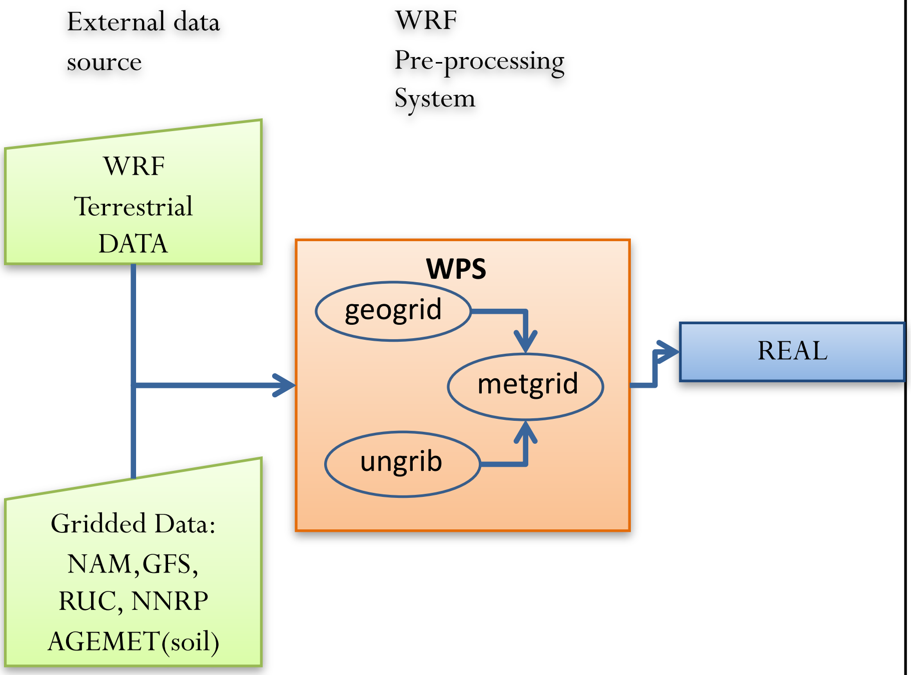
Image: Colin Epperson, Stanford University

WRF domain data



Function of WPS components

- geogrid.exe (think **geo**graphical)
  - Define size/location of coarse domain and interpolate static terrestrial fields to coarse-domain and nested-domain grids
- ungrib.exe
  - Extract meteorological fields from GRIB files
- metgrid.exe (think **met**eorological)
  - Horizontally interpolate meteorological fields (from ungrib) to coarse grid (defined by geogrid)


External data source
WRF Pre-processing System
WRF Terrestrial DATA
Gridded Data: NAM,GFS, RUC, NNRP AGEMET(soil)
WPS
geogrid
metgrid
ungrib
REAL

Developmental Testbed Center

What is GSI?

- GSI is the operational data assimilation system being used at NOAA and NCEP-EMC, and can be applied for global and regional analysis, and can also serve as the observation operators for the Ensemble Kalman Filter (EnKF)
- GSI, started as three dimensional variational data assimilation (3DVar), can be run as a data assimilation system of 2DVar (for surface data analysis), 3DVar, 3D ensemble-variational (3D EnVar), 4D ensemble-variational (4D EnVar), 3D/4D hybrid EnVar, or 4DVar
- For WRF applications, GSI can use the output from real.exe (i.e., wrfinput_d01) as the background field, and update it using the various observations. The updated background field – so called GSI analysis, can then be used as the initial conditions for WRF forecasts
- GSI can also use the WRF forecast files (i.e., wrfout_d01_<yyyymm-dd_hh:mm:ss>) as the background fields and update it for further forecasts.
- The observations can include conventional observations, satellite radiance, GPS radio occultations, etc.

What does GSI do?

GSI background (i.e.,wrfinput_d01)

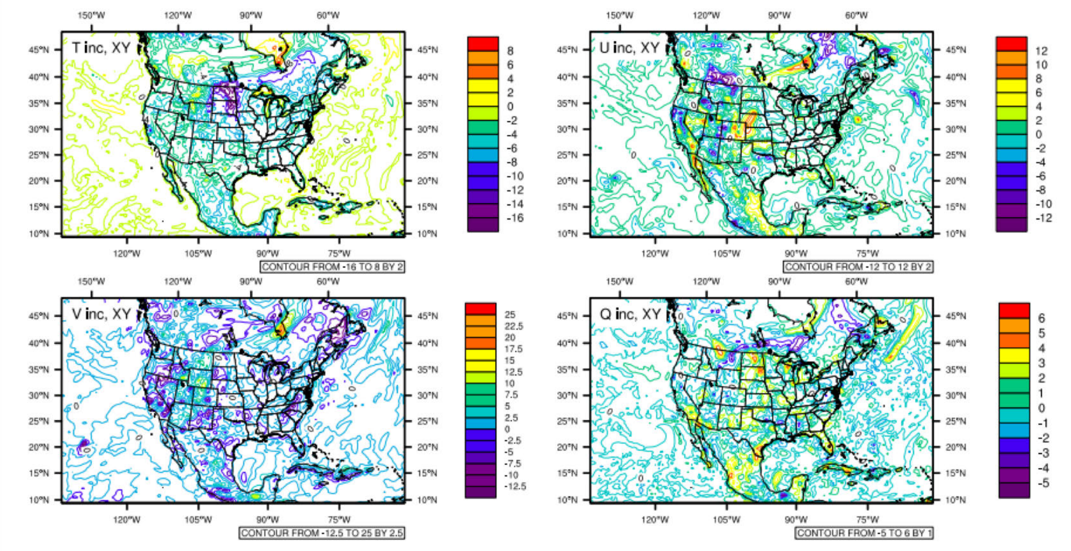
Observations in PrepBUFR/BUFR format (i.e., rap.t12z.prepbufr.tm00

Fix files (background errors, observation errors, CRTM coefficients, data selection, quality control, etc)

GSI executable (gsi.x)

GSI analysis

WRF

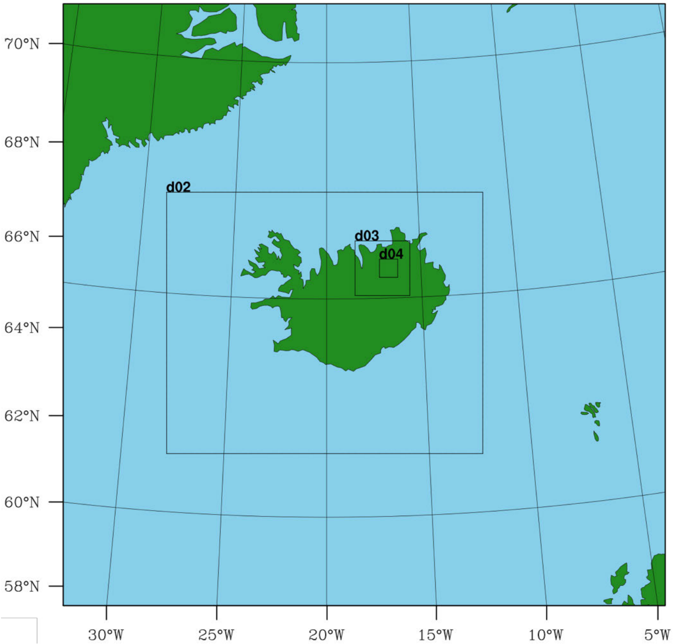


GSI analysis increment (analysis-background) after assimilating conventional observations

What is WRF?

- The Weather Research and Forecasting (WRF) model is a numerical weather prediction model for weather research and forecasting
- Two dynamical cores, both non-hydrostatic
  - Advanced Research WRF (ARW), most commonly used
  - Nonhydrostatic Mesoscale Model (NMM), used in some operational forecasting systems
- Highly configurable, but also caters to less advanced users
  - e.g. 26 different microphysics schemes, 10 surface layer schemes, etc.
  - “Suites” of widely-used and tested scheme combinations are provided for casual users
  - Most options can be easily changed at runtime (no re-compilation required)

Example domain with nests

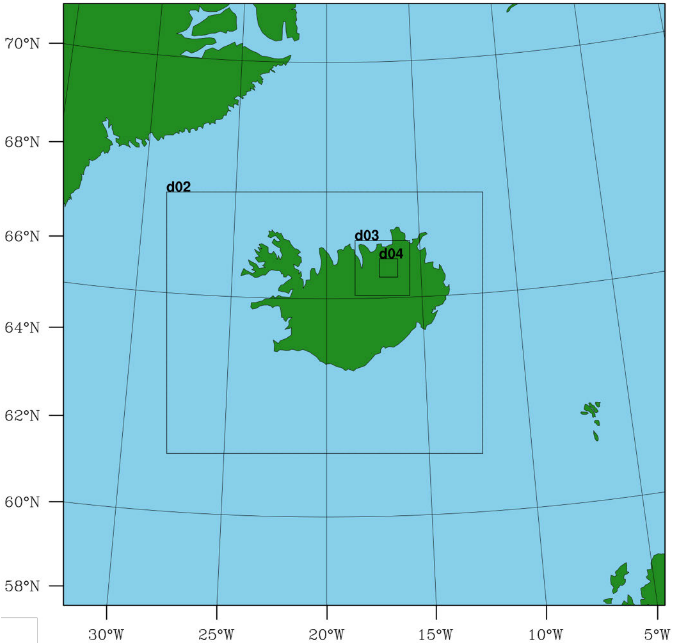


Grid	Resolution	Size
1	16 km	99x99
2	4 km	208x172
3	1 km	136x128
4	250 m	160x160

What is WRF?

- Typically run for regional domains, although a global capability exists
  - One parent domain gets its initial and boundary conditions from the WRF Preprocessing System (WPS)
  - Can also have one or more child domains that get their boundary conditions from the parent domain
- Can be run for specialized applications such as regional climate and atmospheric chemistry (WRF-CHEM) as well

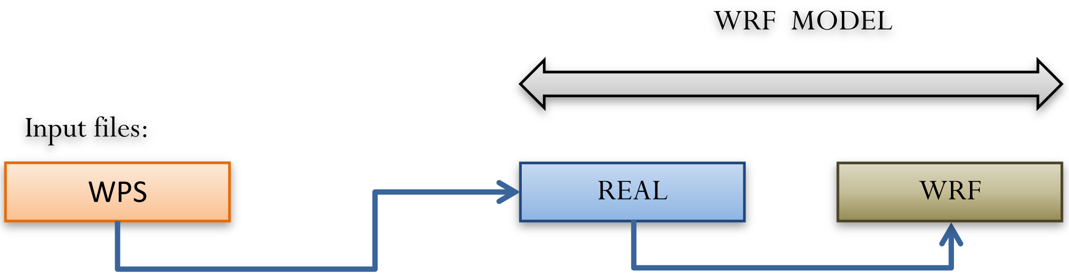
Example domain with nests



Grid	Resolution	Size
1	16 km	99x99
2	4 km	208x172
3	1 km	136x128
4	250 m	160x160

Function of WRF components

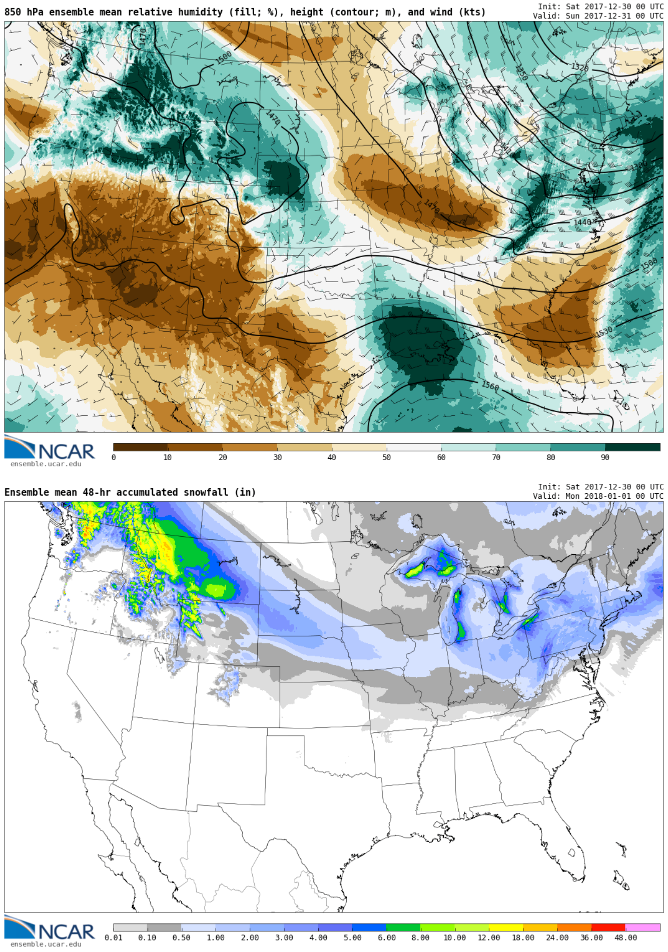
- real.exe
  - Generation of initial state for each of the requested domains
  - Creation of a lateral boundary file for the most coarse domain
  - Vertical interpolation for 3d meteorological fields and for sub-surface soil data
- wrf.exe
  - Forecast model integration through time



What is UPP?

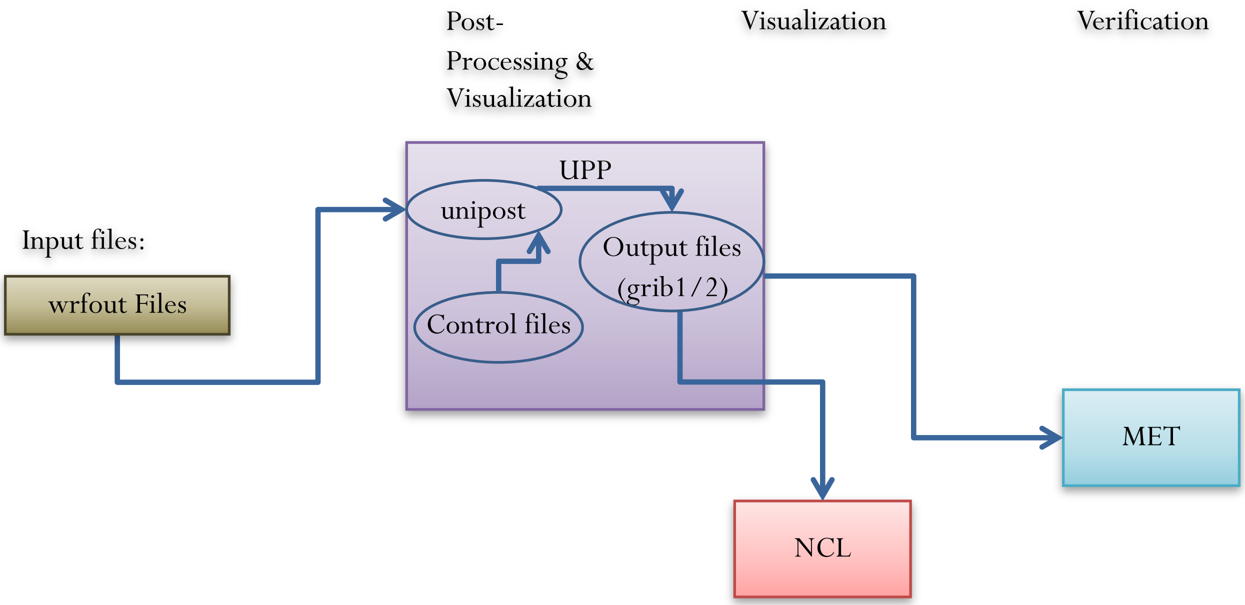
- The Unified Post-Processor (UPP) is a post-processor for WRF and other models
  - Developed at the National Centers for Environmental Prediction (NCEP) for use in its operational forecasting
  - Now offered for community use and development with WRF
- Processes raw model output to more useful forms
  - Produces hundreds of products like those used operationally
  - Creates output that can be plotted with your favorite visualization tool

Example plots from NCAR Ensemble


850 hPa ensemble mean relative humidity (fill; %), height (contour; m), and wind (kts)
Init: Sat 2017-12-30 00 UTC
Valid: Sun 2017-12-31 00 UTC
NCAR
ensemble.ucar.edu
0
10
20
30
40
50
60
70
80
90
Ensemble mean 48-hr accumulated snowfall (in)
Init: Sat 2017-12-30 00 UTC
Valid: Mon 2018-01-01 00 UTC
NCAR
ensemble.ucar.edu
0.01
0.10
0.50
1.00
2.00
3.00
4.00
5.00
6.00
8.00
10.00
12.00
18.00
24.00
36.00
48.00

Function of UPP component

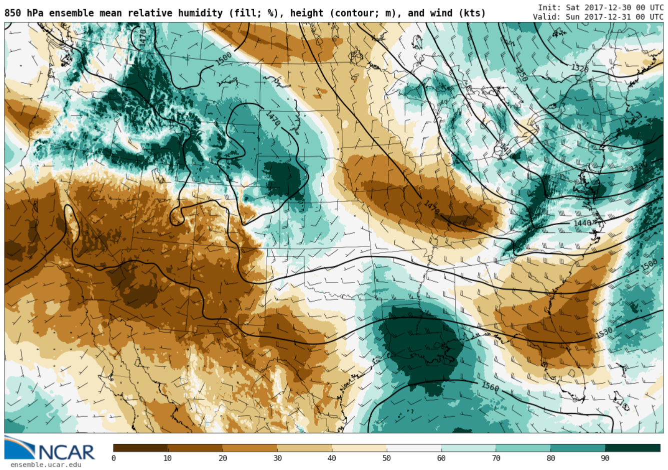
- unipost.exe
  - Performs vertical interpolation from model levels/surfaces onto isobaric, height, and other levels/surfaces
  - Calculated derived quantities/diagnostic fields
  - Destaggers wind onto mass points

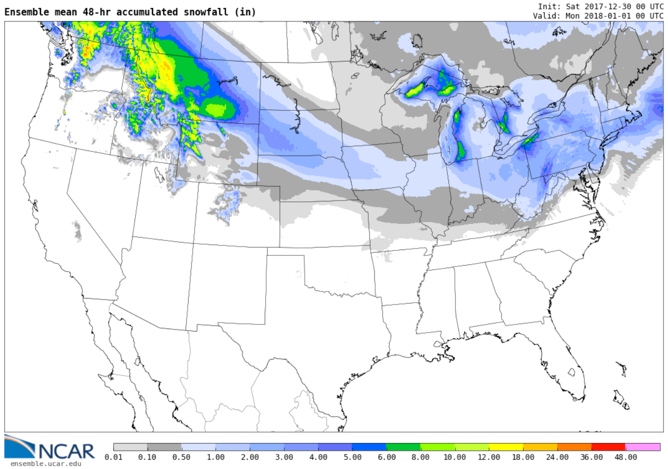


What is NCL?

- The NCAR Command Language
  - An interpreted (scripting) language for creating visualizations of geoscientific data
  - Designed and maintained by the Computational & Information Systems Laboratory (CISL)
  - Can read and write a bunch of different formats (netCDF, HDF4, binary, and ASCII data) and can read many more (HDF5, GRIB1/GRIB2)
  - Supports calling of C and Fortran external routines
  - More info: https://www.ncl.ucar.edu/overview.shtml

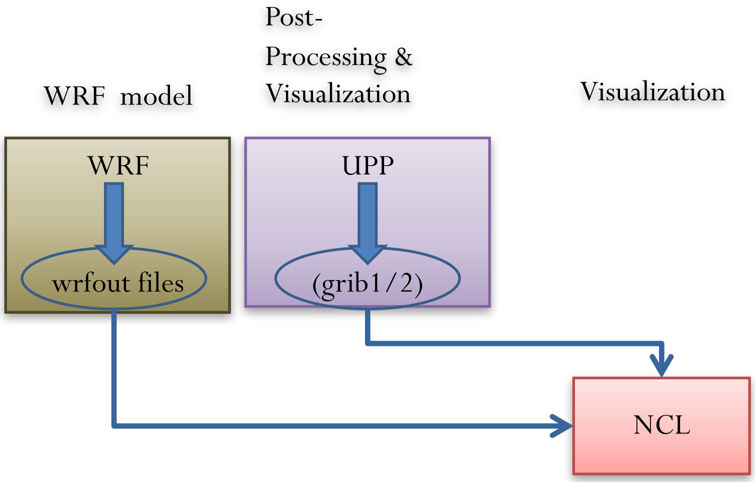
Example plots from NCAR Ensemble


850 hPa ensemble mean relative humidity (fill; %), height (contour; m), and wind (kts)
Init: Sat 2017-12-30 00 UTC
Valid: Sun 2017-12-31 00 UTC
1470
1500
1320
NCAR
ensemble.ucar.edu
0
10
20
30
40
50
60
70
80
90


Ensemble mean 48-hr accumulated snowfall (in)
Init: Sat 2017-12-30 00 UTC
Valid: Mon 2018-01-01 00 UTC
NCAR
ensemble.ucar.edu
0.01
0.10
0.50
1.00
2.00
3.00
4.00
5.00
6.00
8.00
10.00
12.00
18.00
24.00
36.00
48.00

Function of NCL component

- ncl
  - Reads wrfout files created by WRF
    - Calculates derived/accumulated variables (if necessary)
    - Creates plots of one or more variables
  - Can also read grib files created by UPP



What is MET?

- MET is freely available community code supported by the DTC **(must register to download)**
  - State-of-the-art suite of verification tools
  - Approximately 3400+ registered users spanning 125+ countries
  - Users from universities, government, private companies, and non-profits
- Download MET release and compile locally
  - Register and download: **www.dtcenter.org/met/users**
  - C++ with calls to some Fortran libraries, GSL, netCDF4 and HDF5
  - Can be compiled with with GNU, Portland Group (PGI), or Intel
- Support
  - Online tutorial and in-person tutorials given yearly
  - **met_help@ucar.edu** help desk
  - 250+ support tickets in past year

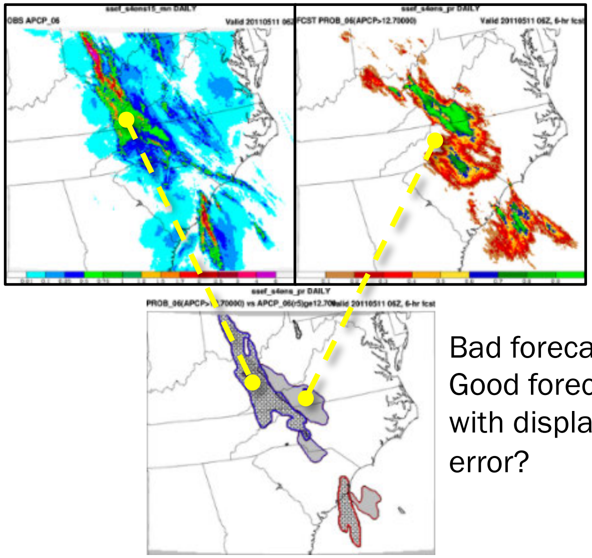
MET
Model Evaluation Tools

A verification toolkit designed for flexible yet systematic evaluation

(supported to the community via the DTC)

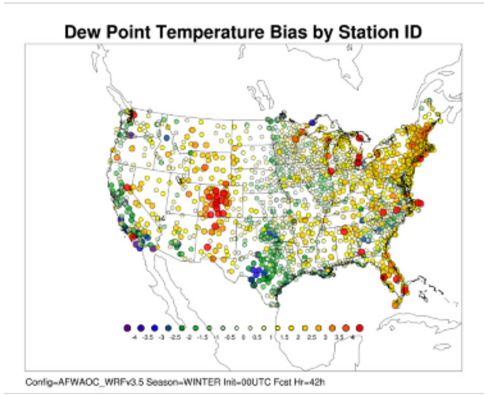
- **Over 70 statistics** using both point and gridded datasets
- Full suite of standard statistics with cutting-edge statistics regularly added
- Computation of confidence intervals
- Able to read in GRIB1, GRIB2 and CF-compliant NetCDF
- Applied to many spatial and temporal scales
- Multiple interpolation methods
- Regridding within the tools and ability to apply complex masking

Object Based and Spatial Methods

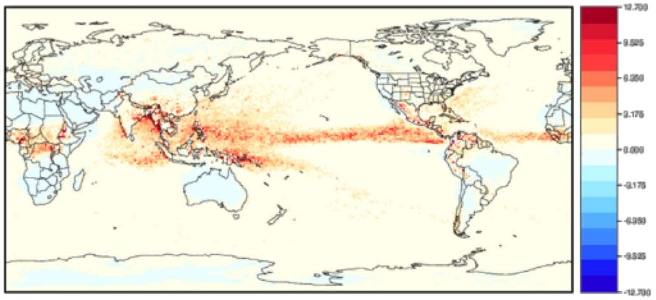

OBS APCP_06
FCST PROB_06(APCP>12.70000)
Valid 20110511 06Z, 6-hr fcst
PROB_06(APCP>12.70000) vs APCP_06(5)ge12.70

Bad forecast or Good forecast with displacement error?

Geographical Representation of Errors


Dew Point Temperature Bias by Station ID
-4 -3.5 -3 -2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5 3 3.5 4
Config=AFWAOC_WRFv3.5 Season=WINTER Init=00UTC Fcst Hr=42h

90th Percentile of difference between two models

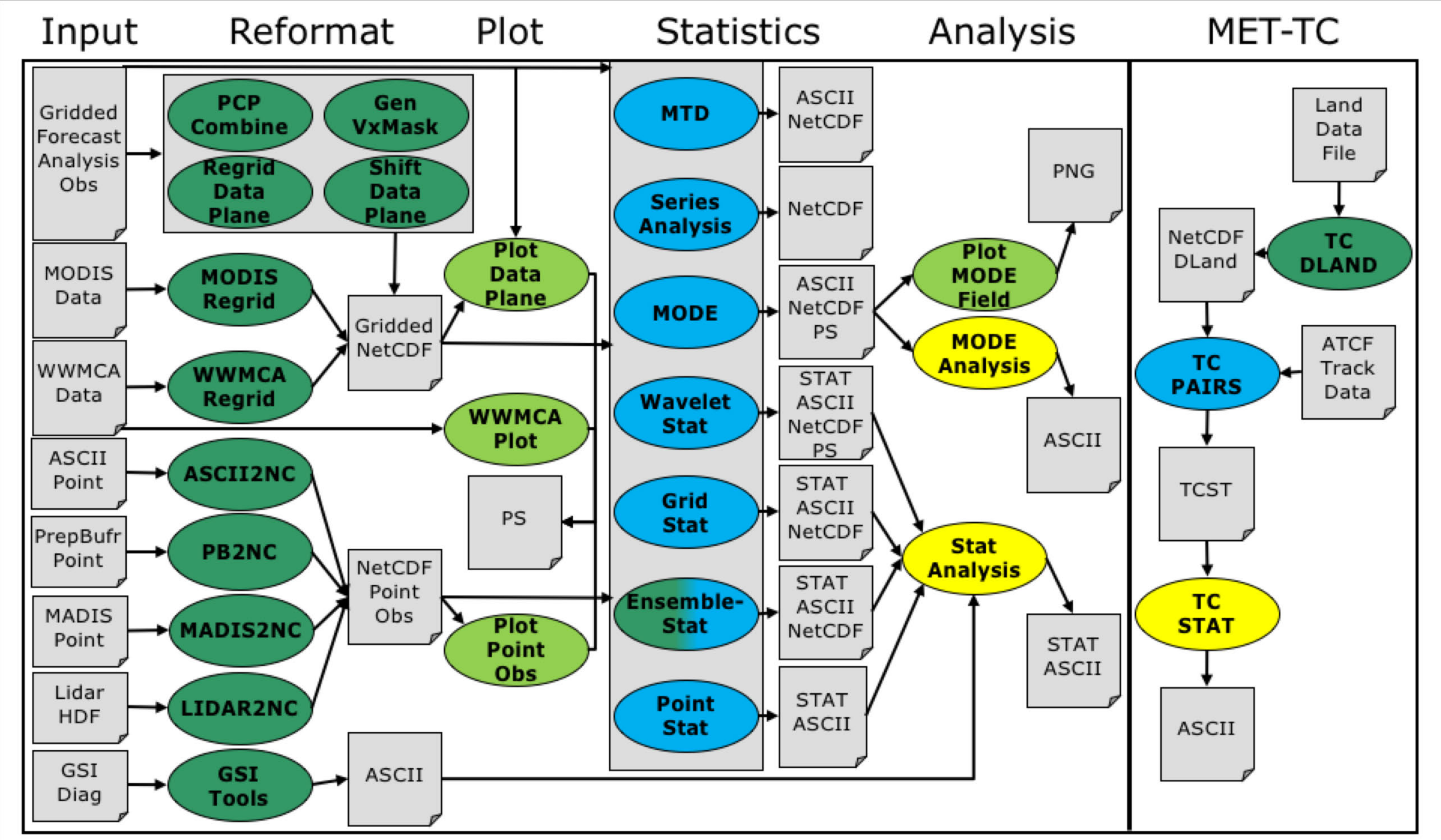


Function of MET

MET includes tools for:

- reformatting (dark green)
- plotting (light green)
- calculating statistics (blue)
- statistical analysis (yellow)
- tropical cyclone verification (MET-TC)

Components Overview



MET data formats & tools

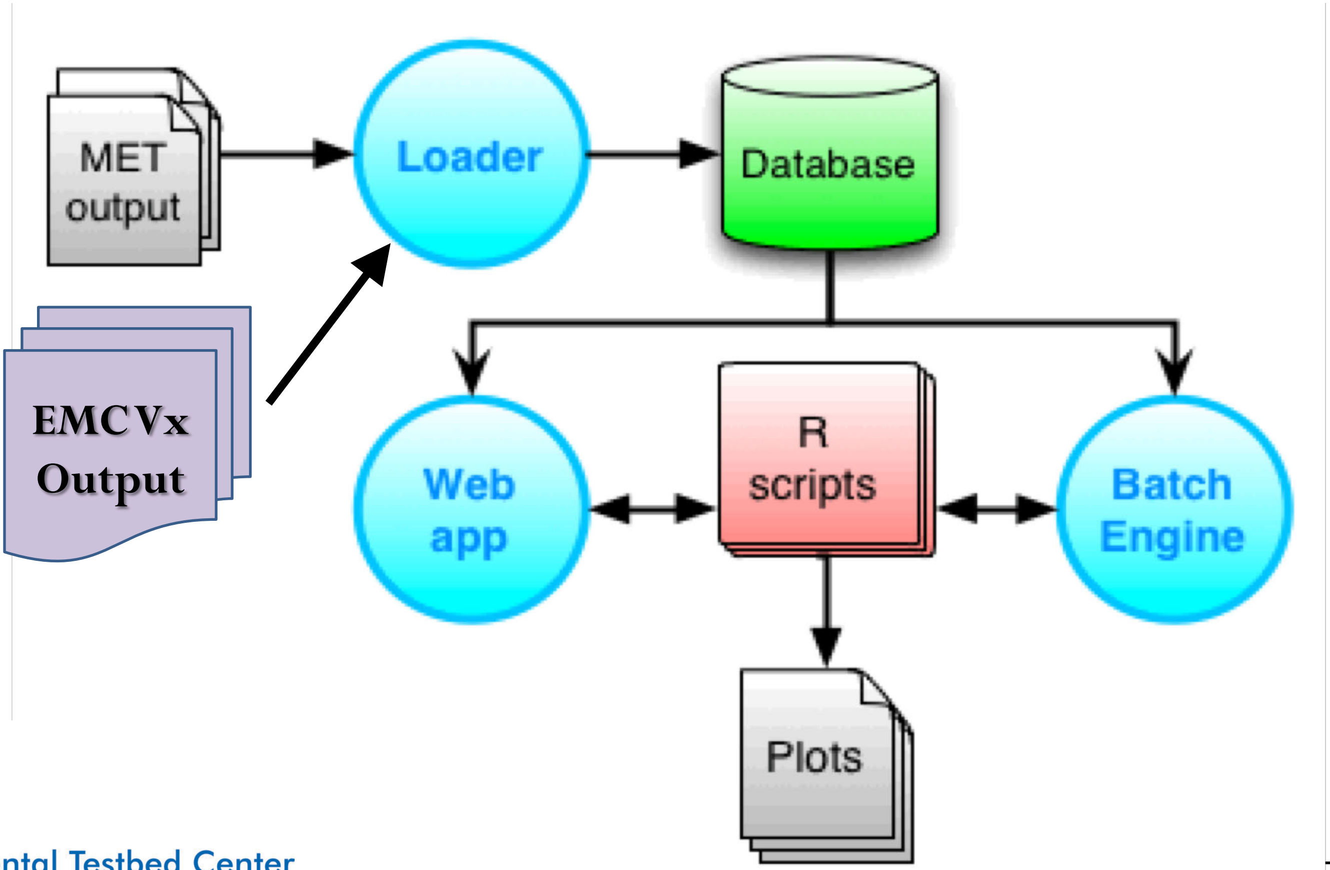
MET components highly-configurable:

- Verify over specified fields and/or levels
- Apply thresholds
- Apply various interpolation methods
- Verify over user-specified regions

Data	MET Tool
Gridded Forecasts Gridded Observations (GRIB1 / GRIB2 / CF-compliant NetCDF / WWMCA / MODIS)	Grid-Stat Ensemble-Stat Wavelet-Stat MODE / MODE-TD Series-Analysis
Gridded Forecasts Point Observations (ASCII formats / Bufr / PrepBufr / MADIS / Lidar)	Point-Stat Ensemble-Stat
Point Forecasts Point Observations (ATCF file format)	TC-Pairs TC-Stat GSI Tools

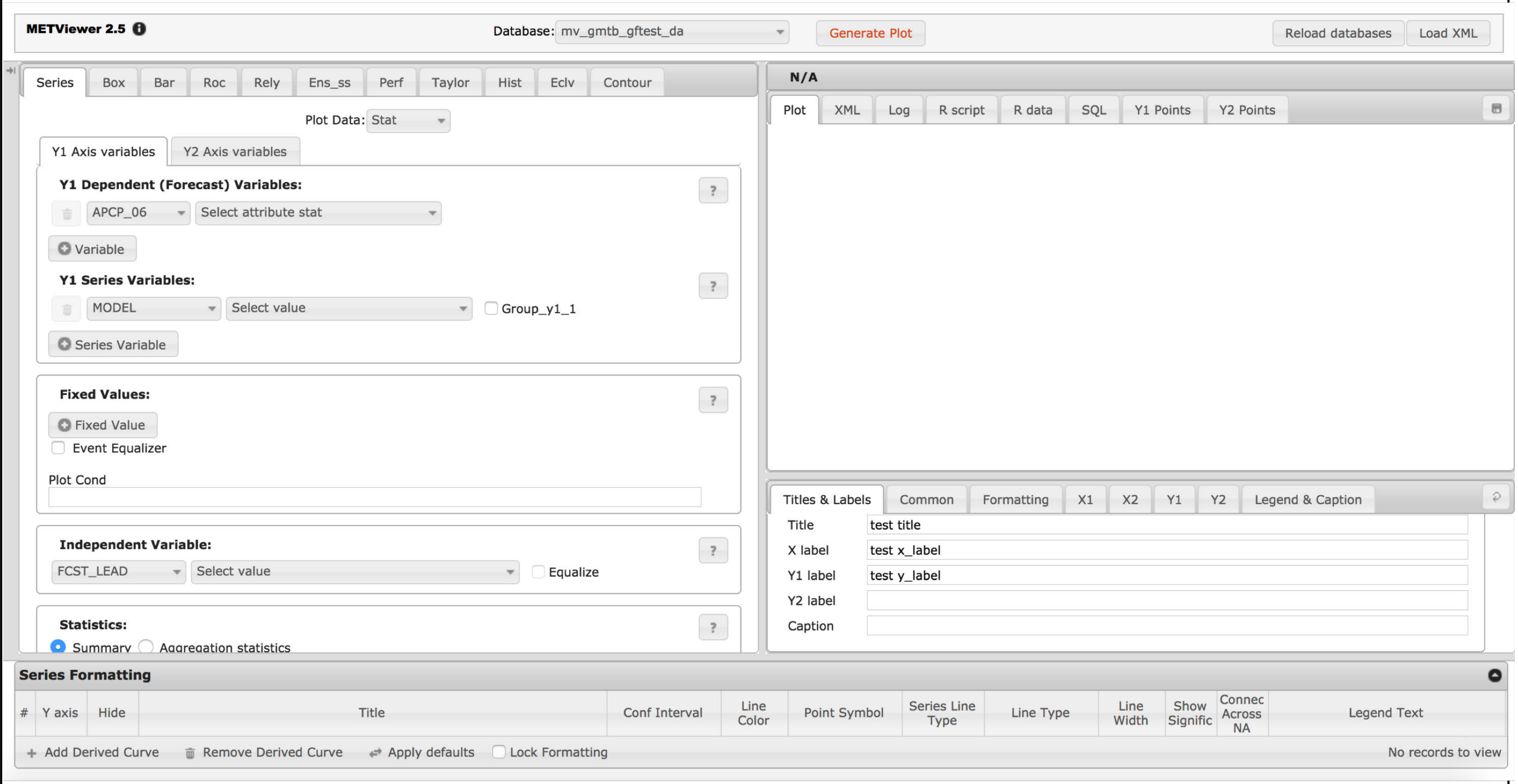
What is METviewer?

- Database and display analysis tool
  - **<u>Packages</u>**: Java, Apache/Tomcat, MySQL, R statistics



METviewer

Customization options



METViewer 2.5

Database: mv_gmtb_gftest_da

Plot Data: Stat

Y1 Dependent (Forecast) Variables:

APCP_06 | Select attribute stat

Y1 Series Variables:

MODEL | Select value | □ Group_y1_1

Fixed Values:

□ Event Equalizer

Plot Cond

Independent Variable:

FCST_LEAD | Select value | □ Equalize

Statistics:

● Summary ○ Aggregation statistics

N/A

Title	test title
X label	test x_label
Y1 label	test y_label
Y2 label	
Caption	

Series Formatting

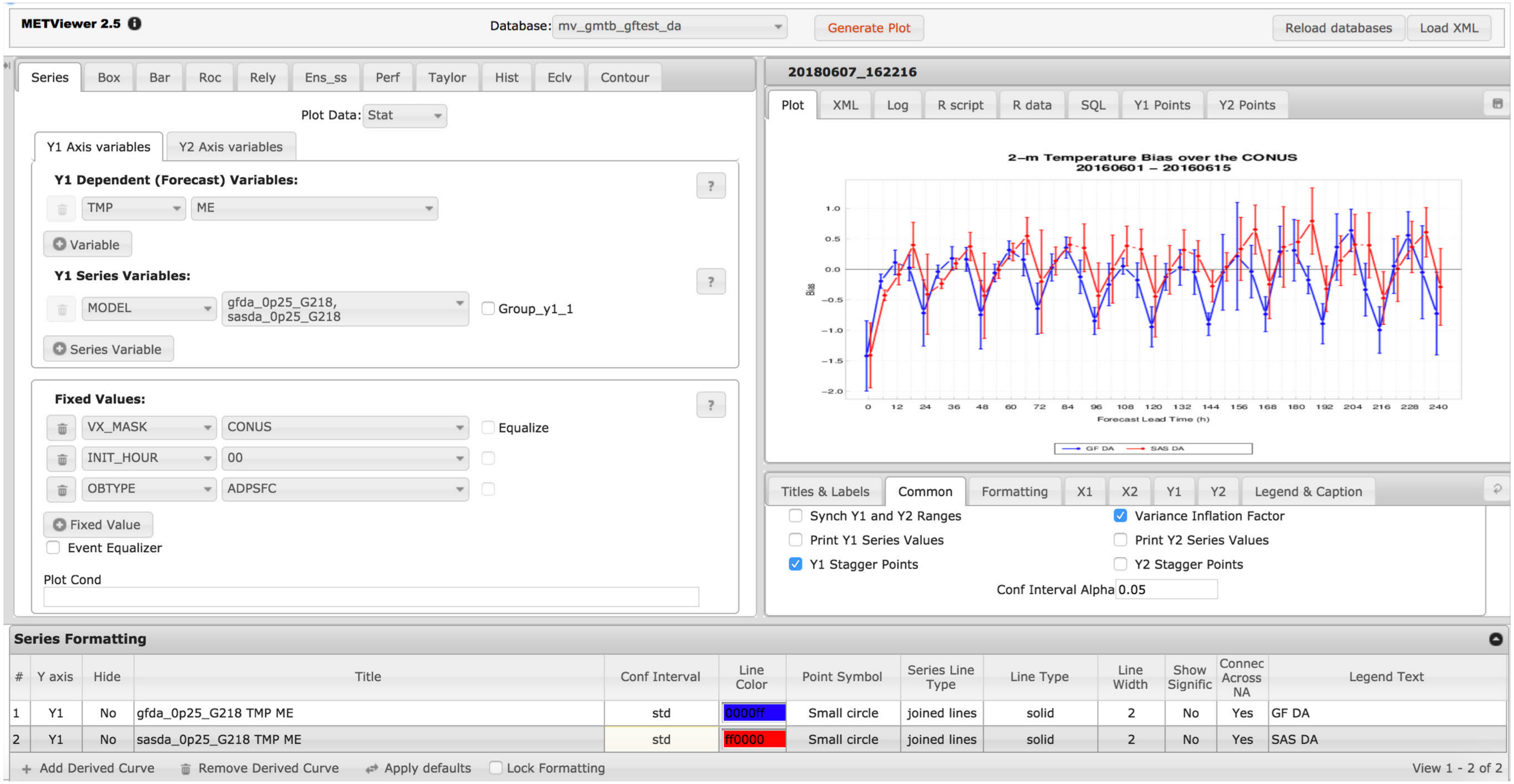
#	Y axis	Hide	Title	Conf Interval	Line Color	Point Symbol	Series Line Type	Line Type	Line Width	Show Signific	Connec Across NA	Legend Text

□ Lock Formatting

No records to view

METviewer

Plotting example



METViewer 2.5 Database: mv_gmtb_gftest_da

Plot Data: Stat

Y1 Dependent (Forecast) Variables:

TMP ME

Y1 Series Variables:

MODEL gfda_0p25_G218, sasda_0p25_G218 □ Group_y1_1

Fixed Values:

VX_MASK CONUS □ Equalize

INIT_HOUR 00 □

OBTYPE ADPSFC □

□ Event Equalizer

Plot Cond

20180607_162216

□ Synch Y1 and Y2 Ranges ☑ Variance Inflation Factor

□ Print Y1 Series Values □ Print Y2 Series Values

☑ Y1 Stagger Points □ Y2 Stagger Points

Conf Interval Alpha 0.05

Series Formatting

#	Y axis	Hide	Title	Conf Interval	Line Color	Point Symbol	Series Line Type	Line Type	Line Width	Show Signific	Connec Across NA	Legend Text
1	Y1	No	gfda_0p25_G218 TMP ME	std	0000ff	Small circle	joined lines	solid	2	No	Yes	GF DA
2	Y1	No	sasda_0p25_G218 TMP ME	std	ff0000	Small circle	joined lines	solid	2	No	Yes	SAS DA

□ Lock Formatting

View 1 - 2 of 2

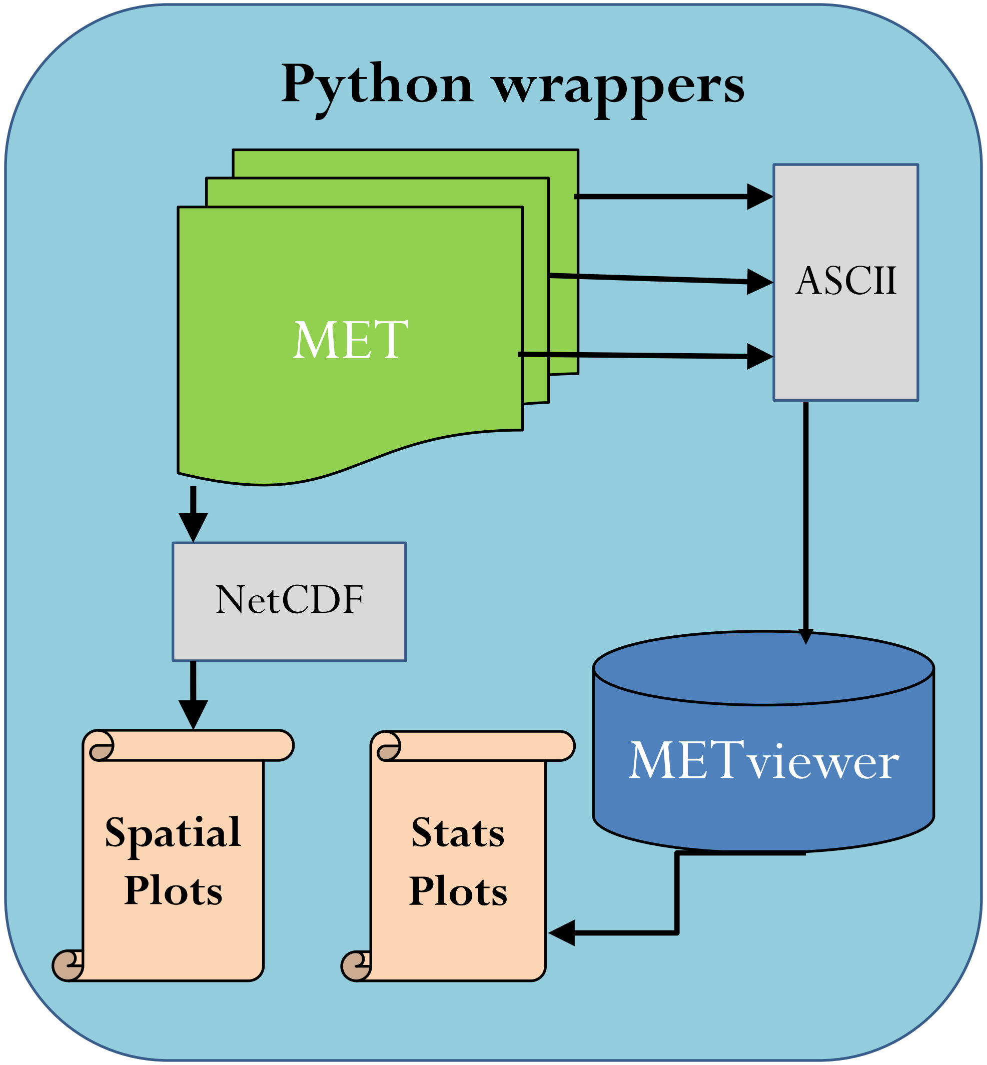
Moving toward METplus

Unified verification package

Available on GitHub:

https://github.com/NCAR/METplus

- Python wrappers around MET and METviewer:
  - Simple to set-up and run
  - Automated plotting of 2D fields and statistics
  - Communication between MET & python algorithms (Cython)
  - Several use cases already established



Software Packages Links

- WPS and WRF
  - Users' Page: http://www2.mmm.ucar.edu/wrf/users/
  - Online Tutorial: http://www2.mmm.ucar.edu/wrf/OnLineTutorial/index.htm
- GSI
  - Users' Page: https://dtcenter.org/com-GSI/users/
  - Online Tutorial: https://dtcenter.org/com-GSI/users/tutorial/index.php
- UPP
  - Users' Page: https://dtcenter.org/upp/users/
  - Online Tutorial: https://dtcenter.org/upp/users/support/online_tutorial/UPPv3.2/index.php
- NCL
  - Users' Page: http://www.ncl.ucar.edu/
- MET
  - Users' Page: https://dtcenter.org/met/users/
  - Online Tutorial: https://dtcenter.org/met/users/support/online_tutorial/index.php