Collaborative WRF-based research and education enabled by software containers

J. Hacker, J. Exby, K. Fossell
National Center for Atmospheric Research

Contributions from Tim See (U. North Dakota)
Why the Weather Research and Forecasting (WRF) model in Docker?

- WRF is a state-of-the-science numerical weather prediction (NWP) model for operations and research
- Compilation and execution can be an intensive effort, slowing time to results
  - Huge complex code
  - Numerous and non-trivial dependencies
  - Inexperienced users can take months to get WRF running for results
- Classroom opportunities for hands-on numerical weather prediction can be intensive to produce
- Research is almost never reproducible
- Collaboration is difficult and cumbersome
Goals of WRF-Docker

• Lower the technical difficulty for new users
  – Graduate students can accomplish early results simultaneous to learning the Unix/Linux skills needed for more in-depth work
  – Provide a reference context

• Trivialize classroom and lab experimentation

• Provide a platform for reproducible numerical weather prediction research

• Facilitate efficient and easy collaboration

WRF in a container is not a black box.
Not a turn-key approach

• Important use cases in an education
  – Change input data sets for land use
  – Change input data sets for initial and boundary conditions
  – Change physics, diffusion, time steps, etc
  – Change code and recompile in known environment

Our container development allows all of these, including deployments on cloud providers or local compute hardware.
Vision: End to end

Acquire data → Set up model (WPS) → Run model (WRF) → Archive metadata → Share output

- CLOUD STORAGE
- LOCAL DISK/SERVER

Any or all on a commercial cloud platform

Run on your command line and link to your filesystem
**Bit-wise reproducibility**

Incomplete list of test platforms

<table>
<thead>
<tr>
<th>OS</th>
<th>Chip/CPU</th>
<th>Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS X 10.10.5</td>
<td>Intel Xeon E5</td>
<td>6</td>
</tr>
<tr>
<td>OS X 10.9.5</td>
<td>Intel Core i7</td>
<td>2</td>
</tr>
<tr>
<td>Ubuntu 14.04</td>
<td>AMD Opteron 6320</td>
<td>16</td>
</tr>
<tr>
<td>Ubuntu 14.04</td>
<td>Intel Xeon E5</td>
<td>16</td>
</tr>
<tr>
<td>RHEL</td>
<td>Intel Xeon X5550</td>
<td>8</td>
</tr>
<tr>
<td>*Ubuntu 14.04</td>
<td>Intel Xeon E5-2666 v3</td>
<td>32</td>
</tr>
<tr>
<td>**Ubuntu 14.04</td>
<td>Intel Atom C2550</td>
<td>4</td>
</tr>
</tbody>
</table>

Reproducible on all platforms so far
University of North Dakota Classroom Implementation

• Access Docker-WRF through Amazon Web Services.
• Students completed a classroom assignments to create an ensemble output of a tornadic supercell over North Dakota.
• Students personally changed the parameterization schemes within WRF.
• Classroom discussion generated through changing of parameterizations.
• Sample Plots below

CAPE

Accumulated Precipitation

Surface Analysis

From Tim See, UND)
University of North Dakota Classroom Implementation

- Total Cost for Homework Assignment: $40.21 over 11 days
  - Inflated cost due to not shutting down instances properly first day.
- Reproducibility of Docker allows for plots to remain the same across all students.
- Students were able to complete their model runs from personal laptops.
Dynamic pull of terrestrial data sets (WPS_GEOG) – work in progress

Current practice

“Local” Disk (desktop, HPC, cloud, etc.)

- Download datasets
- Run geogrid
- Store files locally or remove & repeat when necessary

Downsides:

- Required to download and store entire datasets and unnecessary data
- Not conducive to cloud or container environments
- Large files to store and transfer
- Costs ($) associated with storing or downloading/transferring data in cloud
- Computational inefficiencies in containers due to size of files
Current practice

“Local” Disk (desktop, HPC, cloud, etc.)

- Download datasets
- Run geogrid
- Store files locally or remove & repeat when necessary

New utility

“Local” Disk

WPS_GEOG Subset Utility

- Containerized & Standalone Package
- Reads namelist.wps for defined model grid and desired resolution
- Creates a list of only necessary data files for given domain and resolution
- Downloads subset of files required

Run geogrid (Manual or Container)

Option to keep subset of files locally -or- remove them
Zero to WRF!
in 3 Steps

Using a relatively ”modern” laptop, workstation, server. Linux, Mac, Windows:

• 1. Install docker for free!  https://docker.com/products/

• 2. git clone  https://github.com/NCAR/container-wrf
   (Hurricane Sandy and Katrina)

• 3. cd 3.7.1/demos/local ; docker-compose up
   • (psssst. Windows users- please first edit docker-compose.yml for output DIR)
   
   VOILA!

May 31 2017
So a University professor gives their student a dollar to “do some atmospheric science” in one hour, then write a paper to help toward graduation!

Is this possible?

[Icons representing various activities and tools]
32 core WRF run on AWS

- script to spin up a new AWS resource for our compute access only

- script to launch the docker-compose.yml elements:
  - two containers with data
  - a container with wrf executables
  - a container with NCL scripts to post process
  - copy files from AWS back to macbook

- visualize results on macbook pro.

May 31 2017
Live Demo Time

- Fingers crossed wifi is fast!
Next Steps

• Dynamic data query for smaller downloads

• AWS Batch

• Singularity and Swarms
NCAR Resources


- NCAR Github repo: [https://github.com/NCAR/container-wrf](https://github.com/NCAR/container-wrf)

- NCAR Dockerhub repo: [https://hub.docker.com/r/bigwxwrf/](https://hub.docker.com/r/bigwxwrf/)
  - bigwxwrf/ncar-wrf
  - bigwxwrf/ncar-wpsgeog
  - bigwxwrf/ncar-wrfinputkatrina
  - bigwxwrf/ncar-wrfinputoutsandy
  - bigwxwrf/ncar-ncl
  - bigwxwrf/ncar-wrfinputsandy
  - bigwxwrf/ubc-wrf

- Slack channel for docker-wrf community discussion.
  - email: exby@ucar.edu for invitations
  - [https://ncar-dockerwrf.slack.com](https://ncar-dockerwrf.slack.com)