Cloud-based WRF Downscaling
Simulations at Scale using Community Reanalysis and Climate Datasets

Luke Madaus -- 26 June 2018
luke.madaus@jupiterintel.com

2018 Unidata Users Workshop
Outline

- What is Jupiter?
- Challenges to running WRF at scale
- Jupiter’s approach to this problem
  - Leveraging community datasets
  - Containerization
  - Distributed data-proximate analysis
- (Brief) demonstration
- Example use cases
- Looking ahead to broader access
- Questions for the group
What is Jupiter?
What is Jupiter?
What is Jupiter?

Translating the Latest Climate Science to the Needs of Diverse Users
What is Jupiter?

Publish!

Don’t perish!

Industry

Academia

Governments
What is Jupiter?

Cloud-Native Platform
Data + Models + Analysis

End-Users Interested in Climate Risk
Government + Corporate + Public

Earth System Researchers
Academic + Non-Profit + Industry
What is Jupiter?

FloodScore (Operations and Planning)

HeatScore

How are hyper-local-scale risks for flood and extreme heat hazards changing on daily to decadal timescales?
Regional WRF Simulations on the cloud -- Challenges
Regional WRF Simulation

Existing Community Datasets
- Large-Scale NWP
- Reanalysis
- Climate Simulations

Regional WRF Simulation

Local Value-Added Products
- Case-studies
- Operational NWP
- Model Modification Studies
- Downscaling

JUPITER
Challenges

Existing Community Datasets

- Large-Scale NWP
  - Distributed in netCDF, WRF designed for GRIB!
    - Current Solution: Complicated reformatting and repackaging

- Reanalysis
  - Variables not in the right format!
    - Current Solution: Have to do some sort of calculations or manipulations on data to make it usable for WRF

- Climate Simulations
  - Large files designed for bulk downloads!
    - Current Solution: Have to download everything even if you just need a handful of dates/variables
Challenges

- How do I make WRF work on my system?
  
  Current Solution: While WRF and supporting libraries have gotten easier to compile and install, still a challenge

- Regional WRF Simulation

- I need to move my work to a new cluster. How?
  
  Current Solution: Reinstalling libraries, recompiling, running baseline checks again...

- How do I share all the data I’m generating?
  
  Current Solution: Downloading to local machine, setting up own file server...

- I want to simply swap out another version of my code?
  
  Current Solution: Changing dynamic links, keep track of pointers, making sure things aren’t overwritten...
Regional WRF Simulations on the cloud -- Jupiter’s Approach
Jupiter Platform

- Locally-held Datasets
- Remotely-hosted Datasets

DAP Interface

Containerized Geophysical Models

Cloud-optimized Analysis Platform
- Containerized Postprocessing Suites
- Distributed Interactive Analysis

Amazon Web Services

- Jupiter
- netCDF
- unidata
- Docker
- PANGEA
Jupiter Platform

Locally-held Datasets

Remote-hosted Datasets

DAP Interface

The Data

- netCDF format as current standard
- Increasing number of weather and climate datasets hosted through streamable platforms (e.g., THREDDS)
- On-the-fly subsetting, minimize download size
- Jupiter exploring more cloud-optimized data formats and hosting methods (e.g., Zarr...more on this at the end)
DAP-Fronted Community Datasets

NASA -- CREATE-IP
Reanalysis Datasets (Potter et al. 2018)

Jupiter Platform

The Models

- NSF BigWeatherWeb: Containerization allows confident deployment of models at scale (Hacker et al. 2017)

- Preprocessing automated -- *direct from netCDF to WRF*

- Batch and on-demand simulation for research and real-time forecasts

- Pushing computational limits of current cloud compute architectures

What is containerization?

A portable, self-contained, only what is needed package that can perform some function on any machine running an interface layer.
What is containerization?

Pre-built container images can be “pulled” from online repositories

Or, a single text file that contains a “recipe” for building the container can be downloaded for local builds
Scalable WRF Simulations in <10 lines of Python

- Python-based interface to platform
- Cloud-based storage keeping simulation data organized and sharable
- Simulation “job” options self-contained in Python object
- Explore costs of running simulation with various cloud options
- Command-line submission of configured jobs to cloud compute
Demonstration
Jupiter Platform

The Analysis

- Containerization allows standardized post-processing or analysis chains to be deployed
- Distributed analysis of large datasets through interactive tools (Pangeo stack)
- Allow end-users to dynamically query data for questions of interest
Example Uses

Extensive WRF benchmarking tests across a variety of cloud resource configurations.
Example Uses

Hurricane Hugo (1989) from the perspective of multiple reanalyses

- None of this reanalysis data held locally
- Total pre-processing time: 1 minute per simulation (in parallel)
- WRF execution: 5 day simulation, nesting down to 3 km, 1hr 30min (in parallel)
- Conception to execution -> about 10 lines of Python on Jupiter platform
Example Uses

- Dynamical downscaling simulations to project extreme precipitation in Charleston, SC using 10 member of the CESM-LENS
- Thousands of simulations of extreme precipitation events; completed in one weekend
Hyper-local Flood Risk

Jupiter FloodScore (pilot)

- Metro Region: Charleston
- Variable: Combined Surge
- Scenario: High SLR

Combined Surge
- 1in
- 2in
- 3in
- 4in
- 5in
- 6in
- 1ft
- 2ft
- 3ft
- 4ft
- 5ft
- 6ft

Annual % Chance: 1%

2018
2025
2030
2050
Looking Ahead to Broader Access
NSF Award #1450488
Overview at: http://bigweatherweb.org/Big_Weather_Web/Home/Home.html

BAMS Article: A Containerized Mesoscale Model and Analysis Toolkit to Accelerate Classroom Learning, Collaborative Research, and Uncertainty Quantification

Containerized WRF available!
https://github.com/NCAR/container-wrf
https://hub.docker.com/r/bigwxwrf/ncar-wrf/

Coming soon: Code for generating WRF initial conditions from arbitrary netCDF files
Working on the Jupiter Platform

- Jupiter Containerized WRF targeted for Autumn 2018 release (GitHub)

- Jupiter platform access on a project-by-project basis
  - Let me know if you think you have a good use case!
Questions for this group

1. Cloud-optimized netCDF
   a. An ever-growing need -- case for it is only strengthening
   b. Is this the way to go?
   c. How easy would it be?
2. Containerized WRF simulations
   a. Could you envision using this given your current compute platforms/resources?
   b. What features would you find useful in a standalone, containerized WRF?
3. Cloud-hosted datasets and data-proximate workflows
   a. What would it take to motivate you to move some (or all) of your work to the cloud?

luke.madaus@jupiterintel.com | @lmadaus | http://www.jupiterintel.com