Assignment #1: Compiling and Running WRF-ARW on *comet*

Due: 26 September 2017

This assignment is designed to give you experience with compiling and running the WRF-ARW numerical model on *comet*, the national supercomputer which we will be using for selected course assignments and for the course project. It also serves as a crash course in Linux-based computing. The assignment that you turn in is primarily a curated set of screen captures. Consequently, you can e-mail me your assignment (before the start of class on 26 September) rather than print a copy and turn it in during class.

Prerequisites

1. (5 pts) Register for an XSEDE user account. To do so, go to https://portal.xsede.org and click on the "Create Account" button. Once you have completed registration, please e-mail me with your user name so that I can add you to our resource allocation. *You will not be able to complete any model-related tasks later in this assignment until this portion of the assignment is completed, and so I encourage you to complete this task as soon as possible!*

   After creating your account, on the main XSEDE portal website, click on the "Profile" link in the blue upper menu bar. In the upper right-hand side of this page, click the link to enroll in Duo two-factor authentication. Accept the terms and enroll. *This step is also necessary to be able to complete later tasks in this assignment!*

   For later reference, the XSEDE "Getting Started" guide is available at:

   https://portal.xsede.org/documentation-overview

   Likewise, the *comet* User's Guide is available at:

   https://portal.xsede.org/sdsc-comet

   comet is a supercomputer located at the San Diego Supercomputing Center. All access to *comet* is remote – i.e., you use your computer to securely yet remotely log in to *comet*.

2. (10 pts) Successfully log on to *comet*. To connect to the XSEDE Single Sign-On Hub and then to *comet*, you need a Secure Shell (SSH) client, preferably one that can display remote windows.

   If you are using a Mac, you already have an SSH client available to you via the Terminal. The Macs in the Atmo Lab (W434) and your offices already have the needed software to

display remote windows; however, if you are using a personal Mac, you may need to install the Xquartz package to add this capability to your machine.

If you are using a Windows machine, you must install an SSH client. I recommend using MobaXTerm Home Edition (http://mobaxterm.mobatek.net/download.html). This includes both an SSH client and the ability to display remote windows.

To connect to *comet*, you first log on to the XSEDE Single Sign-On Hub:

ssh -Y <username>@login.xsede.org

where <username> is replaced by your actual username. The above command is what you would type if connecting from the Terminal on a Mac; if using MobaXTerm, click on the "Session" button at upper-left, then on SSH. Enter login.xsede.org in the "Remote Host" entry, check the "Specify username" box, and enter your username in the box to the right. Note that if it prompts you for your password a second time in MobaXTerm, simply cancel out of that window and things should proceed correctly.

Once logged in to the Single Sign-On Hub, you connect to *comet* using the same command no matter whether you are on a Mac or Windows machine:

gsissh -Y comet

This command is entered into the terminal window at the Single Sign-On Hub prompt (e.g., [<username>@ssohub ~]$).

Once you have logged on to *comet*, run the date command by typing date and hitting enter. Take a screenshot of your screen at this point and attach it with your completed assignment.

3. (10 pts) Establish your computing environment for compiling WRF-ARW. All software on *comet*, and many other supercomputers like it, is made available using modules. This way, users can specify only the software they need, and system administrators can simplify the basic computing environment that all users start with.

   To compile and run WRF-ARW, we need to load several modules: the compiler, software that allows us to run the model on many separate processing nodes at once, and software that WRF-ARW uses to read input data and write model output. Some of these are available system-wide, whereas others have been developed for our use only.

   So that these modules are available to you every time that you log in, we place their loading

commands into the .bashrc file in your home directory on *comet*. The .bashrc file contains a list of commands for the supercomputer to load every time you log in. To do so, use the nano text editor:

nano ~/.bashrc

The leading ~/ in front of .bashrc is a shortcut to your home directory. The default .bashrc file contains a few lines that should not be edited that are followed by a comment "# User specifc aliases and functions." Use your keyboard's arrow keys to go below this line and enter the following lines:

```
module unload intel
module load intel/2016.3.210
module load mvapich2_ib
module load hdf5
export HDF5="/opt/hdf5/intel/mvapich2_ib"
module use /oasis/projects/nsf/wim108/evans36/modules/
module load netcdf-local
module load grib2
```

Once you have entered these lines, save the file using Ctrl-O on your keyboard, then exit nano using Ctrl-X on your keyboard. Log off of *comet* back to the XSEDE Single Sign-On Hub, then reconnect to *comet* as you did in Question 2 above.

To test that you completed these commands correctly, after logging back on to *comet*, type ncdump and hit enter. Take a screenshot of your screen at this point and attach it with your completed assignment.

4. (10 pts) Establish links to your scratch and working directories. *comet* is organized around multiple disk storage spaces:

    - Your home directory (/home/<username>). This is a small storage space that is not intended for completing computing tasks.
    - A scratch directory (/oasis/scratch/comet/<username>/temp_project). Your scratch directory has the ability to store large amounts of data (*O*(TB)), but it is a temporary storage space that is shared among all supercomputer users. You should run all of your model simulations in this directory but you should move their output after they complete to your working directory.
    - A working directory (/oasis/projects/nsf/wim108/<username>). This is where you should compile all of your code and complete all of your assignments. While ample

storage is available in this directory, it is shared among your classmates, with ~350 GB available to each person.

Note that your username on comet may, in rare instances, be different than your username used to log on to the Single Sign-On Hub! When logged on to comet, you can check what it is by looking at the text on the command line between the [and @. You should use this username, and not your XSEDE user name, in all instances where <username> appears in the list above and in the questions that follow.

As you might surmise from the last bullet above, your scratch and working directories do not exactly have easy-to-remember names. However, you can make them available from your home directory, with easier-to-remember names, using symbolic links.

Start by changing into your home directory by typing cd and hitting enter. cd is the Linux command for *c*hange *d*irectory; when followed by a directory name, it will change into that directory. Next, create a symbolic link using ln -s:

ln -s full name of directory to link to name of link

Repeat for your other folder. Once this is done, confirm that each were created by listing the contents of your home directory:

ls -al

Take a screenshot of your screen at this point and attach it with your completed assignment. Any time you log on to *comet*, you should make sure you first change directory into either the scratch or working directory, whichever is more applicable for your job, noting that you often will be switching between the two (particularly when running WRF-ARW).

Compiling and Running WRF

The remainder of this assignment follows the WRF-ARW Online Tutorial to guide you through downloading, compiling, configuring, and running the WRF-ARW model. The Online Tutorial is available at:

http://www2.mmm.ucar.edu/wrf/OnLineTutorial/index.htm

You may start by going to the Introduction menu at upper left, then choosing Getting Started.

5. (10 pts) Download and unpack WRF-ARW v3.9 and WPS v3.9.0.1 following the Getting

Started section of the Online Tutorial. A couple of helpful hints:

- Make sure you are connected to *comet* and are in your working directory! You will only need to enter your scratch directory when running the model itself.
- To download a file directly to *comet*, right-click (Windows) or control-click (Mac) on the download link, choose "Copy link address" (Chrome; other browsers should be similar), and move to your terminal window. In the directory where you wish to download the file, type wget followed by a space, then paste the copied address and hit enter. wget is a Linux utility to remotely download files.
- For this and later questions, follow the instructions for Real Data cases only.

Once the model code has been successfully downloaded to and unpacked in your working directory, run ls -al in this directory to list its contents. Take a screenshot of your screen at this point and attach it with your completed assignment.

6. (15 pts) Configure and compile the WRF-ARW model code. Page through the WRF Code section of the Online Tutorial to the Configure WRF section of the tutorial. When presented with the long list of compile options, choose option 15. When asked about nesting, choose option 1. Once this completes, page to the Compile WRF for Real Data Cases section of the tutorial and compile the model code. Note that successful compilation will take 30-60 minutes, so ensure that you are somewhere with a reliable internet connection before you start this question.

   Once done, run ls -al in your test/em_real/ directory to list its contents. Take a screenshot of your screen at this point and attach it with your completed assignment.

7. (10 pts) Configure and compile the WPS code. Whereas WRF-ARW is the model itself, WPS is a set of preprocessing programs that establish the model domain, unpack data to initialize the model, and interpolate it to the desired model domain. Page through the WPS Code section of the Online Tutorial to the Configure WPS section of the tutorial. (Note that you do *not* need to set the NETCDF environment variable as the WPS Code page implies – this is already done by loading the netcdf-local module upon login.)

   When presented with a list of compile options, choose option 19. Page to the Compile WPS section of the Online Tutorial and follow the instructions to compile WPS. (Note that you do not need to compile the utility programs that this page provides as an option.) Once the WPS code has been successfully compiled, run ls -al in the WPS directory to list its contents. Take a screenshot of your screen at this point and attach it with your completed assignment.

At this point, you should read through but not complete any of the tasks listed in the Run Basics, Basics – GEOGRID, Basics – UNGRIB, Basics – METGRID, and Basics – WRF sections of the Online Tutorial. This will give you an overview of the WRF-ARW and WPS workflows, whereas completing the tutorial simulation for the remainder of this assignment will give you experience with actually running WPS and WRF-ARW programs.

8. (5 pts) Create the domain for the January 2000 tutorial case. Page through the January 2000 Case section of the Online Tutorial to the "Set up the model domain" section of the tutorial. Follow the instructions in this section of the tutorial, noting that the needed terrestrial data are already available on *comet* at /oasis/projects/nsf/wim108/evans36/wrfgeog/; this is the path you will enter for geog_data_path in namelist.wps.

   To complete the ncl command listed, you must first load the ncl module:

   module load ncl_ncarg

   Note also that you should use the plotgrids_new.ncl, rather than plotgrids.ncl, script when running ncl. Use this to obtain a graphical display of the model domain, take a screenshot of the resulting window, and attach it with your completed assignment. Upon completing geogrid.exe, run ls -al in your geogrid.exe directory, take a screenshot, and attach it with your completed assignment.

9. (10 pts) Complete the ungrib.exe and metgrid.exe sections of the Online Tutorial. Note that you can use wget to download the case study data tarfile straight to *comet*, as we did for the model code in #5 above. Once you have completed metgrid.exe, use ncview to take a quick look at the met_em.d01.2000-01-24_12:00:00.nc file. ncview is a simply utility that allows you do quickly, yet crudely, view the contents of georeferenced netCDF files, such as those produced by WPS and WRF-ARW.

   To make ncview available, you must first load the ncview module:

   module load ncview

   You can tell ncview to load a particular file by appending the filename to the end of ncview on the terminal line. Once the ncview window opens, display the HGT_M 2D variable. After the display appears, take a screenshot and attach it with your completed assignment. You can quit ncview after this is done.

10. (15 pts) Complete the Run WRF section of the Online Tutorial. To run real.exe and wrf.exe, we use what are known as job submission scripts to submit the code to a scheduler, which

then runs the code when the needed computer resources are available. Sample submission scripts for real.exe and wrf.exe are in /oasis/projects/nsf/wim108/evans36/samplescripts/; you can copy these to your own WRF working directory using cp:

cp /oasis/projects/nsf/wim108/evans36/samplescripts/*.sbatch .

Note that this command assumes that you are in your WRF/test/em_real/ directory. These submission scripts should not need to be edited to run the Tutorial case, though they will need to be edited in later assignments and for your project simulations. More information on the job scheduler is available through the *comet* documentation linked in question #1.

Next, we wish to copy the contents of your WRF/test/em_real/ directory from your working directory to a new directory in your scratch directory. Start by changing into your scratch directory. Create a new directory called WRF_run with mkdir:

mkdir WRF_run

Next, change into your WRF/test/em_real/ directory. Copy the contents of this directory to your scratch directory with cp:

cp * ~/scratch/WRF_run/

The above command would copy the contents of the em_real directory to the scratch/WRF_run directory that can be accessed from your home directory. If you named your scratch directory something other than scratch in #4, make sure that you substitute that name for scratch in your copy command. After you have successfully copied the contents of your em_real directory, change into the WRF_run directory.

When you are ready to run real.exe, submit the real.sbatch script to the job scheduler:

sbatch real.sbatch

To check on its status, use the squeue command:

squeue | grep <username>

where you substitute your username for <username>. squeue provides information about all running and submitted jobs, the | symbol sends the output from squeue to what comes after it, and grep <username> filters the output from squeue to only include lines that have your username in them.

Depending on how much of the supercomputer is being used at a given time, it may take a little while before the code actually runs (i.e., does not say '(Priority)' in the last column for your job). If your code is running, the last three columns will indicate how long it has been running, how many computer nodes it is running on, and the computer nodes on which it is running. If squeue comes back empty, then your submitted job is done. Once done, follow the "verify that the program runs successfully" instructions in the Online Tutorial.

When you are ready to run wrf.exe, submit the wrf.sbatch script to the job scheduler.

Once wrf.exe has completed, use ncview to open the resulting wrfout_d01_2000-01-24_12:00:00 file. Have it load the 4D variable T. By default, it will load T at the lowest model level at the first output time. Change to the last output time by clicking in the box under "Current" for the Time entry in the bottom of the main ncview window until the number in that box matches that in the Max column immediately to its right. Once on the correct time, take a screenshot and attach it with your completed assignment.