



## What is WRF?

The Weather Research and Forecasting Model (WRF) is a state-of-the-art atmospheric modelling system that is applicable for both meteorological research and numerical weather prediction. It offers a host of options for atmospheric physical processes and is suitable for a broad range of applications across scales ranging from tens of metres to the global, including:

- Meteorological investigations
- Real-time numerical weather prediction
- Idealised atmospheric simulations
- Data assimilation studies and development
- Coupling with other earth system models
- Modelling and model use instruction and training

This guide to running WRF provides instructions on how to run a standard WRF test case on the HPC Wales systems.

## Step 1 - Log in

The example used in this guide is configured to run on the Cardiff Skylake *Hawk* cluster. Connect to *hawklogin.cf.ac.uk* with your Supercomputing user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal, thus

```
ssh -l your_username hawklogin.cf.ac.uk
```

The steps below involve typing commands (**in bold font**) in the terminal window.

## Step 2 - Load a WRF module

In common with most of the application guides in this series, we assume at the outset that the module of choice would be selected from those originally available on HPC Wales. Thus a necessary first step would be to gain access to that module set by issuing the commands

```
module purge  
module load hpcw
```

Note that the “module purge” command is required to prevent any unintentional collision with pre-existing modules. Subsequently issuing the module command

```
module avail
```

will provide visibility of and access to the entire set of HPCW modules.

A number of WRF binary packages are available.

- List preinstalled WRF versions:  

```
Module avail wrf
```
- Load your preferred version (version 3.5.1 is used in the first example below):  

```
module load wrf/3.5.1
```

- Confirm the loaded modules:

```
module list
```

Note: Dependent modules, e.g. compiler and MPI library, NetCDF, HDF5, szip, etc. are loaded automatically.

### Step 3 - Create run directories

Given the size of data files associated with WRF, we suggest you create two directories for processing WRF jobs— one for the associated run scripts and "small" output plus log files, the second to house the large input data sets associated with the actual computations. The former should be sited in your home directory, the latter in your /scratch directory - /scratch/\$USER. Thus for this tutorial, a directory called WRF should be created (i) in your home directory, thus

```
cd ~
mkdir WRF
cd WRF
```

and (ii) create the following directory under your scratch allocation:

```
mkdir -p /scratch/$USER/WRF/inputFiles
```

The general approach below is to copy the SLURM job script to **\$HOME/WRF**, and the large input files to **/scratch/\$USER/WRF/inputFiles**.

### Step 4 - Obtain a test case

A test case for WRF – "conus12km" [continental US with 12 km resolution] is provided with the installation at:

```
/app/environment/wrf/3.5.1/example/
```

Copy the input files to **/scratch/\$USER/WRF/inputFiles** and the SLURM job script plus reference output files to **\$HOME/WRF**, thus:

```
cd /scratch/$USER/WRF/inputFiles
cp -r /app/environment/wrf/3.5.1/example/inputFiles/conus12km .
```

and

```
cd ~/WRF
cp /app/environment/wrf/3.5.1/example/validation/* .
```

The SLURM run script - run\_wrf.conus12km.SLURM.q – should be used to run the test case, a 3-hour simulation of continental US with 12 km resolution.

### Step 5 - Submit a parallel job

The supplied job script runs using 120 processes with 40 processes per node, and should run for about one minute.

- Submit the job using:
 

```
sbatch run_wrf.conus12km.SLURM.q
```
- Check the job queue using:

## queue

- Note the approach adopted in the job script – to avoid copying the input files to the users scratch space for each run. This takes time & space and is avoided by using the unix “ln” utility to link the input files when running the job.
- When completed, output can be found in a file called WRF.conus12km.o.<JobID> and errors, if any, can be found in WRF.conus12km.e.<JobID> (where <JobID> is the ID generated by the generated by the SLURM queuing software).
- If your job is successful, its output can be found in the directory

```
~/WRF/WRF.conus12km.Hawk.n120.ppn=40.rsl.error.0000.id-<JobID>
```

- If you wish, you can compare your output to the reference output in the directory `/app/environment/wrf/3.5.1/example/validation/WRF.conus12km.Hawk.n120.ppn=40.rsl.error.0000.24495`

## Step 6 - Rerun with other parallel settings

Now Open the `run_wrf.conus12km.SLURM.q` using your favourite editor, such as nano, emacs or vi.

Line 5 of the jobscript reads:

```
#SBATCH --ntasks=120    # number of parallel processes (tasks)
```

meaning that 120 cores are used to run this job. Change this number to, say, 80 and rerun the test case.

Note: The Skylake Gold 6148 clusters have 40 cores per node while the older Sandy Bridge clusters have 16 cores per node. To reduce the core count on a node, the line

```
#SBATCH --ntasks-per-node=40    # tasks to run per node
```

should be changed to:

```
#SBATCH --ntasks-per-node=16    # tasks to run per node
```

Carry out the above reduction – what effect does this have on the overall run time?

## Step 7 - Additional Benchmark Cases

Two additional benchmark cases are described below:

- **CONUS 2.5km** is a 63 million cell benchmark<sup>1</sup>, based on the final 3 hours of a 9-hour, 2.5 km resolution case covering the Continental United States domain on 4 June 2005 with a 15 second time step. The benchmark period is hours 6–9 (3 hours), starting from a restart file from the end of the initial 6 hour period.
- **4dbasic** is from the European WRF Benchmark Suite<sup>2</sup>, and is intended to represent typical user needs in complicated and demanding multi-nest and multi-

---

<sup>1</sup> [http://vsc.ac.at/fileadmin/user\\_upload/vsc/documents/Workshop\\_Neusiedl/Schicker.pdf](http://vsc.ac.at/fileadmin/user_upload/vsc/documents/Workshop_Neusiedl/Schicker.pdf) ; [http://www.nasa.gov/pdf/518437main\\_SAP5\\_1\\_first\\_draft-numbered-061507.pdf](http://www.nasa.gov/pdf/518437main_SAP5_1_first_draft-numbered-061507.pdf)

<sup>2</sup> <http://weather.arsc.edu/WRFBenchmarking/4dbasic.html>

shaped configurations used in regional modelling applications. The nesting is described in [2].

## The WRF module

Note that these two cases have not as yet been validated using the module used for the conus12km case. In these cases use should be made of version 3.4.1, thus

```
module load wrf/3.4.1
```

- Confirm the loaded modules:

```
module list
```

## Create run directories

We suggest using the same approach as in the simpler case above, creating two directories for processing the WRF jobs– one for the associated run scripts and "small" output plus log files, the second to house the large input data sets associated with the actual computations. The former ( $\$HOME/WRF$ ) should be sited in your home directory, the latter in your /scratch directory - /scratch/ $\$USER/WRF$ . Thus for the basic4d example, a directory called **WRF** should be created in your home directory, while using the

```
/scratch/ $\$USER/WRF/inputFiles$ 
```

directory under your scratch allocation. Again you should copy the appropriate SLURM job script to  $\$HOME/WRF$ , and the large input files to `/scratch/ $\$USER/WRF/inputFiles$` .

## Obtaining the Benchmark cases

The benchmark cases for WRF – "conus\_2.5km" and "4dbasic" (see above) are provided with the installation at:

```
/app/environment/wrf/3.4.1/benchmarks
```

Copy the input files to `/scratch/ $\$USER/WRF/inputFiles$`  and the SLURM job script plus reference output files to  $\$HOME/WRF$ , thus:

```
cd /scratch/ $\$USER/WRF/inputFiles$ 
```

```
cp -r /app/environment/wrf/3.4.1/benchmarks/inputFiles/conus_2.5km .
```

```
cp -r /app/environment/wrf/3.4.1/benchmarks/inputFiles/4dbasic .
```

and

```
cd ~/WRF
```

```
cp /app/environment/wrf/3.4.1/benchmarks/validation/* .
```

The SLURM run script - `run_wrf.conus2.5km.SLURM.q` – should be used to run the benchmark conus\_2.5km case, and the `run_wrf.4dbasic.SLURM.q` used to run the 4dbasic case.

## Submit the benchmark parallel jobs

The conus\_2.5km job scripts run using 128 processes with 32 processes per node, while the 4dbasic script uses 256 processors also with 32 per node, although both will scale to much

higher core counts. The conus\_2.5km benchmark should run for about 17 minutes, the 4dbasic benchmark for 33 minutes.

- Submit the jobs using:  
`sbatch run_wrf.conus2.5km.SLURM.q` and  
`sbatch run_wrf.4dbasic.SLURM.q`
- Check the job queue using:  
`squeue`
- When completed, the output can be found in files called WRF.conus2.5km.o.<JobID> and WRF.4dbasic.o.<JobID>, while the errors, if any, can be found in WRF.conus2.5km.e.<JobID> and WRF.4dbasic.e.<JobID>, (where <JobID> is the ID generated by the generated by the SLURM queuing software).

## Further info

Further information on WRF can be found at the website

<http://www2.mmm.ucar.edu/wrf/users/>, with a list of the available documentation at <http://www2.mmm.ucar.edu/wrf/users/pub-doc.html>