wrf

The Weather Research and Forecasting (WRF) Model is a next-generation mesoscale numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs. It features multiple dynamical cores, a 3-dimensional variational (3DVAR) data assimilation system, and a software architecture allowing for computational parallelism and system extensibility. WRF is suitable for a broad spectrum of applications across scales ranging from meters to thousands of kilometers.

Usage

Most users will wish to build WRF from source so we provide here just a few modifications that you may need to make to build WRF. If you are a researcher at an Australian university you should follow the particular instructions at this wiki.

The WRF ARW online Tutorial is essential reading first. The following instructions may be out of date so it is best to refer to the Wiki given above.

Assuming the source code has been downloaded from http://wrf-model.org/index.php and that you wish to build the distributed memory version with the Intel compilers then you need the following modules loaded:

```
intel-cc, intel-fc, openmpi, netcdf/3.6.3
```

Also make sure that the environment variable NETCDF is set to /apps/netcdf/3.6.3. At the configure step chose option 7 (Intel compilers and DMP).

Then edit configure.wrf to have

```
LIB_EXTERNAL    = $(WRF_SRC_ROOT_DIR)/external/io_netcdf/libwrfio_nf.a  \
```

Note the extra directory Intel added as well as -lnetcdff.


Once you have run compile wrf you should end up with the executable wrf.exe in the rundirectory. An example job script to run a WRF job would be:

```
#!/bin/csh
#PBS -l mem=8Gb,ncpus=8,walltime=01:00:00
#PBS -l wd
mpirun -np $PBS_NCPUS wrf.exe
```

It is possible to build the shared memory (SMP) version of WRF and also a hybrid (DMP + SMP) version. If you build the shared memory version you are restricted to running on a maximum of 8 cpus, the size of the shared memory nodes of the vu. If you wish to use the hybrid version please read the man page for mpirun to see the options that are required. It is likely that the adequate performance and scaling can be obtained with the DMP version.

Hints from the WRF web site:

- If you are running WRF model with OpenMP (smpar compile) and you have set something like "unlimit" or "ulimit -s unlimited", or set MP_STACK_SIZE to 64000000 or larger, the model still crashes, this may be a problem of over specifying stack size. Solution: Set stack size sufficiently large, but not unlimited value. A few GB is usually more than enough. If your computer or compiler does not recognize MP_STACK_SIZE, try KMP_STACKSIZE and/or OMP_STACKSIZE.
- When using the CAM scheme (option 3) for the long and shortwave radiation physics (options are set the namelist.input file) you need to add two new compiler options to FCBASEOPTS_NO_G in configure.wrf so that you have

```
FCBASEOPTS_NO_G = -w -ftz -align all -fno-alias -no-ip -fno-inline -fp-model precise $(FORMAT_FREE) \\
$(BYTESWAPIO)
```

Then clean out the object files by doing make clean then recompile (compile em_real to generate real.exe and wrf.exe.

Building WPS with grib support

Users must define two environment variables before configuring WRF or WPS.

```
setenv JASPERINC /usr/include
setenv JASPERLIB /usr/lib64
```

The WRF_LIB lines in the configure script will need -lnetcdff added as above.
Building WPS with Intel Compilers

If you use ncl.6.0.0 you will need to add -lgfortran to the link command as NCL is built with the GNU compilers. Linking with the libraries in ncl/5.2-Intel does not require this extra library.