Running global WRF on Odyssey (V1.0, 02/12/2017)

If you don't have any experiences with WRF, it is strongly suggested to go through the official online tutorial (http://www2.mmm.ucar.edu/wrf/OnLineTutorial/index.htm) before reading this document. The official tutorial is very informative and helpful.

It would be helpful to go through at least these two parts of the official tutorial:
1, Get an overview:
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Introduction/start.htm
2, One simple case:
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/CASES/SingleDomain/index.html

If there are any issues, please feel free to reach me at dingma3277@gmail.com. Comments and feedbacks are also welcome and will be greatly appreciated.

The following steps are customized instructions to run global WRF on Odyssey.

Step 1. Collect everything (WRF, WPS and input data), and create your own working directories

1.1 WRF can be found here: /n/home10/kuang/Model/WRFV3/

1.2 WPS can be found here: /n/home09/dingma/Model2/WPSv3/

Note that to successfully compile WPS, WPS must be copied to the same directory where you put WRF.

1.3 Input data:

geog: /n/kuanglfs/dingma/WRF_2014/wrf_input/geog/

reanalysis: either FNL or ncep data can be used as input. Here ncep will be used as an example, which has been downloaded here: /n/kuanglfs/dingma/WRF_2014/wrf_input/data/ncep2_data/

1.4 Environment: put the following lines in your .bashrc file

```
export NETCDF=/n/sw/fasrcsw/apps/MPI/intel/15.0.0-fasrc01/impi/5.1.2.150-fasrc01/netcdf/4.1.3-fasrc09/
export PATH=$PATH:/n/sw/fasrcsw/apps/MPI/intel/15.0.0-fasrc01/impi/5.1.2.150-fasrc01/netcdf/4.1.3-fasrc09/bin/
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export WRF_EM_CORE=1
export WRF_NMM_CORE=0
export WRF_CHEM=0
export WRF_KPP=0
```

1.5 Modules needed:

```
source new-modules.sh
module purge
module load intel/15.0.0-fasrc01
module load impi/5.1.2.150-fasrc01
module load libpng/1.6.25-fasrc01
module load jasper/1.900.1-fasrc02
module load netcdf/4.1.3-fasrc09
```

Step 2. Compile WRF

2.1 Go to the folder to which you copied WRFV3.

2.2 Clean the source codes: ./clean –a

2.3 Configure the compilation: ./configure

2.3.1 Enter selection 7 for platform: "Linux x86_64 i486 i586 i686, ifort compiler with icc (non-SGI installations) (dmpar)"

2.3.2 Enter selection 1 for nesting: "1=basic"

2.3.3 Compile: ./compile em_real &> compile.log &

2.3.4 Grab some coffee. If you really want to watch it compiling: ./tail –f compile.log

2.3.5 If it goes well, it will take around 30 minutes or so, and it will create wrf.exe and real.exe in YOUR_WORKING_DIR/WRFV3/main

Step 3. Compile WPS

3.1 Go to the folder to which you copied WPS.

3.2 Clean the source codes: ./clean –a

3.3 Configure the compilation: ./configure

Enter selection of your choice (No. 1 would normally work) for platform.

3.4 Compile: ./compile &> compile.log &

Step 4. Generate input data.

4.1 A namelist can be found here:
/n/home10/dingma/Model2/WPSv3/monsoon_160km/namelist.wps

4.2 Edit your namelist accordingly, and follow the steps as in the official tutorial (./ungrib.exe, ./geogrid.exe, and ./metgrid.exe). The simple but useful example is here: http://www2.mmm.ucar.edu/wrf/OnLineTutorial/CASES/SingleDomain/index.html

4.3 Note that if you are using ncep data, you will have to unpack both pressure level (PGB) data and surface data (FLX).

Step 5. Generate initial/boundary conditions.

5.1 A namelist can be found here:
/n/home10/dingma/holylfs/WRF_2016/real/monsoon_160km/namelist.input

5.2 Edit your namelist accordingly, and follow the step as in the official tutorial (./real.exe).

Step 6. Run the simulation.

6.1 A namelist can be found here:
/n/home10/dingma/holylfs/WRF_2016/RUN/monsoon_160km/namelist.input

6.2 Edit your namelist accordingly, and submit the simulation (./sbatch job_wrf).
An example of the submission file can be found here:
/n/home10/dingma/holylfs/WRF_2016/RUN/monsoon_160km/job_wrf