Compilation steps of models and necessary libraries

1. Load the intel MPI library

You have to load the intel MPI library

```
$ source /opt/intel/compilers_and_libraries_2017/linux/mpi/bin64/mpivars.sh
```

Use the command “which” to check if the intel MPI commands are loaded properly. Check with the version of the library.

```
$ which mpif90
/opt/intel/compilers_and_libraries_2017.2.174/linux/mpi/intel64/bin/mpif90

$ which mpicc
/opt/intel/compilers_and_libraries_2017.2.174/linux/mpi/intel64/bin/mpicc

$ which mpirun
/opt/intel/compilers_and_libraries_2017.2.174/linux/mpi/intel64/bin/mpirun

$ mpif90 --version
GNU Fortran (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39)
…
```

2. Compilation of the necessary libraries (optional)

NOTE: This session discusses about the compilation of necessaries libraries. You can directly load the necessaries at /r2/local

2.1 Compilation of zlib, hdf5 and netcdf4 libraries

The compilation of WRF requires netcdf4 which in turn requires zlib and hdf5. If you are preferring the latest version of netcdf4 and hdf5, you may download the tar ball and compile your own versions of these libraries.

First, create a “Downloads” directory under your home directory.

```
$ mkdir -p $HOME/Downloads
```

2.1.1 Compilation of zlib (Optional)

Get the latest version of tar ball from the official page at https://zlib.net/. Alternatively copy our provided tar ball at /r2/lib_src/zlib

```
$ cp -r /r2/lib_src/zlib ~/Downloads
$ cd ~/Downloads/zlib
$ tar -xvf zlib-1.2.11.tar.gz
```

Uncompress the tar ball, compile the source code and finally install the libraries to the target local directory under your home. The “prefix” option determines the destination path to be $HOME/local

```
$ cd zlib-1.2.11
$ ./configure --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

2.1.2 Compilation of hdf5 (Optional)

The hdf5 library requires zlib. Thus, zlib has to be installed properly beforehand. Get the latest version of tar ball from the official page at https://www.hdfgroup.org/downloads/hdf5/source-code/. Alternatively copy our provided tar ball at /r2/lib_src/hdf5

```
$ cp -r /r2/lib_src/hdf5 ~/Downloads
$ cd ~/Downloads/hdf5
$ tar -xvf hdf5-1.12.0.tar.gz
```

Uncompress the tar ball, compile the source code and finally install the libraries to the target local directory under home. **NOTE, hdf5 depends on zlib, so zlib has to be built beforehand.**

```
$ cd hdf5-1.12.0
$ ./configure --with-zlib=$HOME/local --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

2.1.3 Compilation of netcdf4 for C and netcdf4 for Fortran (Optional)

The netcdf4 library requires hdf5 and zlib. Thus, they have to be installed properly beforehand. Get the latest version of tar ball from the official page at https://www.unidata.ucar.edu/downloads/netcdf/. Alternatively copy our provided tar ball at /r2/lib_src/netcdf4

```
$ cp -r /r2/lib_src/netcdf4 ~/Downloads
$ cd ~/Downloads/netcdf4
$ tar -xvf netcdf-c-4.7.4.tar.gz
```

Uncompress the tar ball for netcdf4 for C, compile the source code and finally move the libraries to the target local directory. **NOTE** that you have to **set 3 environment variables (LD_LIBRARY_PATH, CPPFLAGS and LDFLAGS)** before configuring:

```
$ cd netcdf-c-4.7.4
$ export LD_LIBRARY_PATH=-L/$HOME/local/lib
$ export CPPFLAGS="-I$HOME/local/include -I$HOME/local/include"
$ export LDFLAGS="-L$HOME/local/lib -L$HOME/local/lib"
$ ./configure --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

Similarly for compiling netcdf4 for Fortran:

```
$ cd ~/Downloads/netcdf4
$ tar -xvf netcdf-fortran-4.5.3.tar.gz
$ cd netcdf-fortran-4.5.3
$ ./configure --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

In order for the correct version of netcdf libraries to be loaded, 2 environment variables have to setup to the .bashrc at home directory. Add the following 2 lines to the end of .bashrc

```
export PATH="$HOME/local/bin":$PATH
export LD_LIBRARY_PATH="$HOME/local/lib":"/usr/lib64":$LD_LIBRARY_PATH
```

Load the settings by sourcing the .bashrc and check the version of our compiled netcdf and see if the paths are correct.

```
$ source ~/.bashrc
$ nf-config --all
```

2.1.4 Compilation of jasper

Get the latest version of tar ball from the official page at https://www.ece.uvic.ca/~frodo/jasper/ . Alternatively copy our provided tar ball at /r2/lib_src/jasper

```
$ cp -r /r2/lib_src/jasper ~/Downloads
$ cd ~/Downloads/jasper
$ tar -xvf jasper-1.900.1.tar.gz
$ cd jasper-1.900.1
$ ./configure --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

2.1.5 Compilation of libpng

Get the latest version of tar ball from the official page at http://www.libpng.org/pub/png/libpng.html . Alternatively copy our provided tar ball at /r2/lib_src/libpng

```
$ cp -r /r2/lib_src/jasper ~/Downloads
$ cd ~/Downloads/jasper
$ tar -xvf jasper-1.900.1.tar.gz
$ cd jasper-1.900.1
$ ./configure --prefix=$HOME/local |& tee log.conf
$ make |& tee log.make
$ make install |& tee log.makeinstall
```

3. Compilation of the model

3.1 Clone the source code of WRF and WPS

The latest version of WRF and WPS can be obtained from the official repository at github.

```
$ cd $HOME
$ mkdir model
$ cd model
$ git clone https://github.com/wrf-model/WRF.git
...
$ git clone https://github.com/wrf-model/WPS.git
...
$ ls
WPS  WRF
```

Alternatively, you can get the V3 source code tar ball at /r2/model/V3.9.1.1/src_tarball

```
$ mkdir -p $HOME/model
$ cd $HOME/model
$ cp -r /r2/model/V3.9.1.1/src_tarball ./
$ cd src_tarball
$ tar -xvf WRFV3.9.1.1.tar.gz; mv WRFV3-3.9.1.1 WRFV3
$ tar -xvf WPSv3.9.1.tar.gz; mv WPS-3.9.1 WPS
$ ls
WPS  WRFV3
```

3.2 Setting up Environment Variables

NOTE: Assume you have all libraries installed under your home directory (i.e. $HOME/local)

Before the compilation of WRF and WPS, **several environment variables need to be setup**. Add the following lines at the end of the file ".bashrc" in your home directory.

```
export CC=gcc
export FC=gfortran
export CXX=g++
export F77=gfortran
export F90=gfortran
export PATH="$HOME/local/bin":$PATH
export LD_LIBRARY_PATH="$HOME/local/lib":"/usr/lib64":$LD_LIBRARY_PATH
export JASPERLIB=$HOME/local/lib
export JASPERINC=$HOME/local/include
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export NETCDF=$HOME/local
```

You must properly set these environment variables before model compilation by sourcing the .bashrc file:

```
$ source ~/.bashrc
```

3.3 Compilation of WRF (version 3)

NOTE: assume you have all models source codes are installed under your home directory (i.e. $HOME/model)

First you have to clean the source code folder:

```
$ cd $HOME/model/WRFV3
$ ./clean -a
```

Then, configure the compilation settings:

```
$ ./configure
```

Select a suitable setting (here option 34 for compiler GNU compilers). Afterwards, there is a file called **configure.wrf**

Then, you can compile the real case simulation for WRF. If no error occurs, should have the following printout.

```
$ ./compile em_real |& tee log.compile
…
==========================================================================
build started:   Sun Nov 29 13:23:19 HKT 2020
build completed: Sun Nov 29 13:40:16 HKT 2020

--->                  Executables successfully built                  <---

-rwxr-xr-x 1 r2 vhpc-hko-r2 40251072 Nov 29 13:40 main/ndown.exe
-rwxr-xr-x 1 r2 vhpc-hko-r2 40103512 Nov 29 13:40 main/real.exe
-rwxr-xr-x 1 r2 vhpc-hko-r2 39641024 Nov 29 13:40 main/tc.exe
-rwxr-xr-x 1 r2 vhpc-hko-r2 44415376 Nov 29 13:38 main/wrf.exe

==========================================================================
```

Finally, you can check if there are any missing dependencies in all the 4 exe files (**ndown.exe, real.exe, tc.exe, wrf.exe**) under the main directory with the command "ldd".

```
$ cd main
$ ls *.exe
ndown.exe  real.exe  tc.exe  wrf.exe
$ ldd wrf.exe
…
```

3.4 Compilation of WPS

NOTE: assume you have all models source codes are installed under your home directory (i.e. $HOME/model)

Since the compilation of WPS needs the compiled files from WRFV3, WRFV3 has to be compiled beforehand. First you have to clean the source code folder and configure the compilation settings.

```
$ cd $HOME/model/WPS
$ ./clean -a
$ ./configure
```

Select a suitable setting (here option 3 for compiler GNU compilers). Afterwards, there is a file called **configure.wps**

Then, you can compile the exe files of WPS

```
$ ./compile |& tee log.compile
```

Finally, you can check if there are any missing dependencies in all the 3 exe files(**geogrid.exe, metgrid.exe, ungrib.exe**) using the command “ldd”

```
$ ls *.exe
geogrid.exe  metgrid.exe  ungrib.exe
$ ldd geogrid.exe
…
```

The End