Documentation of the chemistry-transport model

Chimere

[version 2017r4]

July 25, 2018.

How to install required libraries under GNU/Linux

Contents

1 pNetCDF and NetCDF4 formats

1.1 Problems with NetCDF4 files

pNetCDF is not able to open netcdf4 files as explained in this website: https://trac.mcs.anl.gov/projects/parallel-netcdf
and with the following text on their site:
"NetCDF gives scientific programmers a self-describing and portable means for storing data. However, prior to version 4, netCDF does so in a serial manner.
NetCDF started to support parallel I/O from version 4, whose parallel I/O feature was at first built on top of parallel HDF5. Thus, the file format required by NetCDF-4 parallel I/O operations was restricted to HDF5 format. Starting from the release of 4.1, NetCDF has also incorporated PnetCDF library to enable parallel I/O operations on files in classic formats (CDF-1 and 2). Official support for the CDF-5 format started in the release of NetCDF 4.4.0.
Note NetCDF now can be built with PnetCDF as its sole parallel I/O mechanism by using command-line option "–disable-netcdf-4 –enable-pnetcdf". Certainly, NetCDF can also be built with both PnetCDF and PHDF5 enabled. In this case, a NetCDF program can choose either PnetCDF or Parallel HDF5 to carry out the parallel I/O by adding NC_MPIIO or NC_NETCDF4 respectively to the file open/create flag argument when calling API nc_create_par or nc_open_par. **When using PnetCDF underneath, the files must be in the classic formats (CDF-1/2/5). Similarly for HDF5, the files must be in the HDF5 format (aka NetCDF-4 format).** *NetCDF-4 example programs are available to demonstrate such parallel I/O operations."*
There are two types of parallel I/O with recent netcdf versions:

- Native parallelism with netcdf4 files: not yet supported by CHIMERE
- Parallelism using pnetcdf for classic files. pnetcdf does not support netcdf4 format.

1.2 How to use CHIMERE with already prepared NetCDF4 files?

1. Check the format of the NetCDF file:
   example:

   ```
   ncdump -k geog_USGS_DCWAc2.nc
   classic
   ncdump -k LANDUSE_GLOBCOVER_DCWAc2.nc
   64-bit offset
   ```

   If the result is "classic" or "64-bit offset", CHIMERE with pNetCDF will read easily the files.

2. If this is not the case, you have to convert the netcdf4 files to netcdf3 with:

   ```
   ncks -3 netcdf4_file.nc netcdf3_file.nc
   ```

   For more details on NCO command, the documentation is here: http://nco.sourceforge.net/nco.pdf

1.3 In case of use of WRF

WRF is able to produce NetCDF3 format file directly. If you are using it as meteorological driver, it is necessary to add a specific line in the namelist.input file. This line has to be in the first "block" called *["time_control"]*.

```
&time_control
use_netcdf_classic                  = .true.
```

2 Libraries to install

The CHIMERE model has been tested on GNU/Linux systems with Open MPI message passing libraries. The model requires several numerical tools, compilers and libraries. The system has to installed with GNU **bash** Bourne shell, **awk** and **make**. It is also necessary to have python libraries. Note that **ALL libraries must be compiled with the same compiler as the CHIMERE model.** The model was developed and tested using the softwares versions as described in Tableau 1.

Software	URL	version
Fortran compilers		
gfortran	http://gcc.gnu.org/wiki/GFortran	devtoolset-1.1
ifort	https://software.intel.com/en-us/fortran-compilers	composer_xe_2013
Librairies		
Unidata NetCDF	http://www.unidata.ucar.edu/	netcdf-c-4.3.1
pNetCDF	http://trac.mcs.anl.gov/projects/parallel-netcdf	pnetcdf-1.4.1
Open MPI	http://www.open-mpi.org	openmpi-1.6.5

Table 1: *URL adresses for the development and the use of the CHIMERE model and its modules*

If no **bash** is installed, the model may work with a baseline Bourne-shell, but the user may have to edit the scripts to take in account some syntactic features specific to **bash** . The same remark apply to the **awk** and **make** utilities.
In the following examples, the **wget** utility is used for downloading, because it is a robust and powerful downloading tool. However, if **wget** is not installed on your system, you can obviously use your favorite browser to download files.
The model has been tested with **gfortran** and **intel** compilers. If these compilers are not installed by default on your computer, install one of them to be able to compile the NetCDF libraries and the CHIMERE model.

2.1 Installation of MPI

Depending on your choice, gfortran or ifort, this section presents how to install MPI. We consider here that all computers are now under 64 bit systems.

2.1.1 With gfortran on a 64 bit system

```
export COMPILO=gfortran
./configure CC=gcc CXX=g++ F77=$COMPILO FC=$COMPILO \
             CFLAGS=-m64 CXXFLAGS=-m64 FFLAGS=-m64 FCFLAGS=-m64 \
            --prefix=/opt/openmpi-1.2.5-${COMPILO} \
            --with-mpi-f90-size=medium
make all
make install
```

Edit /etc/profile to set the PATH variable
Edit /etc/ld.so.conf.d/openmpi.conf

2.1.2 With ifort on a 64 bit system

```
export CC=icc
export CXX=icpc
export CFLAGS='-O2 -m64'
export CXXFLAGS='-O2 -m64'
export LDFLAGS='-O2'
export FC=ifort
export FCFLAGS='-O2 -m64'
export F77=ifort
export FFLAGS='-O2 -m64'
./configure --prefix=/opt/openmpi-1.2.5-ifort-64 --with-mpi-f90-size=medium
```

2.2 Installation of NetCDF and pNetCDF

After the download of the NetCDF packages, untar and unzip the files as:

```
tar xvfz netcdf.tar.gz
tar xvfz parallel-netcdf.tar.gz
```

As an example for the NetCDF-3 family version 3.6.3:

```
cd netcdf-3.6.3/src
```

The configuration process depends on several environment variables that you shall set according to your compiler. Although it may be possible to link CHIMERE executables compiled with compiler A with the NetCDF library compiled with compiler B, we strongly recommend to use the same compiler. The time you will spend to install a new NetCDF library is negligible compared to the time you could loose trying to match inconsistent libraries.
We suppose you want to install netcdf somewhere under `/opt` . If you choose to install under `/usr/local`, or under you home directory, replace `/opt` your installation path in the following text.
We also suppose your shell is like `bash` or `ksh` . If you use `csh`, modify the `export` statements according to your shell syntax.

2.3 Compilation with intel

As root :

```
mkdir /opt/netcdf-3.6.3-ifort
```

As an ordinary user :

```
export FC=ifort
export F90=ifort
export CFLAGS="-O -m64"
export FFLAGS="-mp"
export CPPFLAGS="-DNDEBUG -DpgiFortran"
./configure --prefix=/opt/netcdf-3.6.3-ifort
make
make test
```

As root :

```
make install
```

Parallel NetCDF:

```
export LD_LIBRARY_PATH=/opt/openmpi-1.10.2-ifort/lib:/opt/openmpi-1.10.2-ifort/lib/openmpi

export FC=ifort
export F90=ifort

# The same C/C++ compilers as for Open MPI!
export CC=icc
export CXX=icpc
export MPICC=/opt/openmpi-1.10.2-ifort/bin/mpicc
export MPICXX=/opt/openmpi-1.10.2-ifort/bin/mpicxx
export MPIF77=/opt/openmpi-1.10.2-ifort/bin/mpif77
export MPIF90=/opt/openmpi-1.10.2-ifort/bin/mpif90
export CFLAGS='-O -m64 -mcmodel=medium'
export CXXFLAGS='-O -m64 -mcmodel=medium'
export FFLAGS='-O -g -m64 -mcmodel=medium'
export FCLAGS='-O -g -m64 -mcmodel=medium'
export LDFLAGS='-O -g -mcmodel=medium'
./configure --prefix=/opt/pnetcdf-1.7.0-ifort-64-medium
make clean && make && make testing
```

As root:

```
make install
```

2.4 Compilation with gfortran

```
export CC=/opt/centos/devtoolset-1.1/root/usr/bin/gcc
export CXX=/opt/centos/devtoolset-1.1/root/usr/bin/g++
export FC=/opt/centos/devtoolset-1.1/root/usr/bin/gfortran
export F77=/opt/centos/devtoolset-1.1/root/usr/bin/gfortran

export CFLAGS='-O -m64 -mcmodel=medium'
export LDFLAGS='-O -g -fno-second-underscore -mcmodel=medium'
export CPPFLAGS='-DNDEBUG -DgFortran'
export FCFLAGS='-O -g -m64 -fno-second-underscore -mcmodel=medium'
export FFLAGS='-O -g -m64 -fno-second-underscore -mcmodel=medium'

./configure --prefix=/opt/netcdf-3.6.3-gfortran-64-medium
make cleancode&& make && make check && make install
```

As root:

```
  make install
```

Parallel NetCDF:

```
export FC=/opt/centos/devtoolset-1.1/root/usr/bin/gfortran
export F90=/opt/centos/devtoolset-1.1/root/usr/bin/gfortran
export CC=/opt/centos/devtoolset-1.1/root/usr/bin/gcc
export CXX=/opt/centos/devtoolset-1.1/root/usr/bin/g++

export MPICC=/opt/openmpi-1.10.2-gfortran/bin/mpicc
export MPICXX=/opt/openmpi-1.10.2-gfortran/bin/mpicxx
export MPIF77=/opt/openmpi-1.10.2-gfortran/bin/mpif77
export MPIF90=/opt/openmpi-1.10.2-gfortran/bin/mpif90
export CFLAGS='-O -m64 -mcmodel=medium'
export CXXFLAGS='-O -m64 -mcmodel=medium'
export FFLAGS='-O -g -m64 -fno-second-underscore -mcmodel=medium'
export FCLAGS='-O -g -m64 -fno-second-underscore -mcmodel=medium'
export LDFLAGS='-O -g -fno-second-underscore -mcmodel=medium'

./configure --prefix=/opt/pnetcdf-1.7.0-64-gfortran-medium
make clean && make && make testing
```

As root :

```
  make install
```