1. build process

build.sh:

```
#!/usr/bin/bash


export OPTFLAGS="-march=native -Ofast -ffast-math -freciprocal-math
-mprefer-vector-width=256 -mtune=native -flto -ftree-vectorize -fopenmp -mfma
-mavx2 -mfma -m3dnow"
export CC="mpicc -fPIC"
export CXX="mpicxx -fPIC"
export FC="mpif90 -fPIC"
export MPIFC="mpif90 -fPIC"

spack load openmpi@5.0.3/ccc6ge6
spack load libxml2

mkdir -p build
export DEST=`pwd`/build
export
LIBXML2=/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/libxml2-2.10.
3-runp2abl3ecyveymhcmfhwdl6c4y7f54/

./prereq_install.sh


export PATH=$DEST/bin:$PATH
export LD_LIBRARY_PATH=$DEST/lib:$LD_LIBRARY_PATH

export FCFLAGS="-fno-range-check"

export CFLAGS="$CFLAGS $OPTFLAGS"
export CPPFLAGS="$CPPFLAGS $OPTFLAGS"
export LDFLAGS="$LDFLAGS -L$LIBXML2/lib -L$DEST/lib
-Wl,-rpath,$LIBXML2/lib:$DEST/lib"
autoreconf -f -i
./configure --prefix=$DEST --enable-parallel-nc --enable-pnetcdf --enable-bdw
--enable-nhl --enable-nc4-gzip --enable-cdf5
make clean
make version
make install -j `nproc`
```

prereq_install.sh:

```
#!/bin/bash
#
# INSTALL NETCDF and MPI2 LIBRARY IN DEST.
#
# Requires wget program
#
# CHEK HERE BELOW THE COMPILERS
#
# Working CC Compiler
#CC="gcc -fPIC"
#CC="icc -fPIC"
#CC="pgcc -fpic -DpgiFortran"
#CC="xlc_r -qpic"
# Working C++ Compiler
#CXX="g++ -fPIC"
#CXX="icpc -fPIC"
#CXX="pgCC -fpic"
#CXX="xlc++_r -qpic"
# Working Fortran Compiler
#FC="gfortran -fPIC"
#FC="ifort -fPIC"
#FC="pgf90 -fpic"
#FC="xlf2003_r -qpic"
# Destination directory
#DEST=$PWD

pnetcdf_ver=1.12.3
netcdf_c_ver=4.9.2
netcdf_f_ver=4.6.1
hdf5_ver=1.14.1-2
zlib_ver=1.2.13
ompi_ver=4.1.5
ompi_major=`echo $ompi_ver | cut -d "." -f 1-2`
hdf5_major=`echo $hdf5_ver | cut -d "." -f 1-2`

UNIDATA=https://downloads.unidata.ucar.edu/
OPENMPI=http://www.open-mpi.org/software/ompi/v${ompi_major}/downloads
HDFGROUP=https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-${hdf5_major}/hdf5-`echo ${hdf5_ver} | cut -d "-" -f1`/src
ZLIB=https://github.com/madler/zlib/releases/download/v1.2.13
```

```
LIBXML2=/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/libxml2-2.10.3-runp2abl3ecyveymhcmfhwdl6c4y7f54/

export LD_LIBRARY_PATH=$DEST/lib:$LD_LIBRARY_PATH

export CFLAGS="$CFLAGS $OPTFLAGS"
export CPPFLAGS="$CPPFLAGS $OPTFLAGS"
export FCFLAGS="$FCFLAGS $OPTFLAGS"
export FFLAGS="$FFLAGS $OPTFLAGS"

if [ -z "$DEST" ]
then
 echo "SCRIPT TO INSTALL NETCDF V5 and MPICH LIBRARIES."
 echo "EDIT ME TO DEFINE DEST, CC AND FC VARIABLE"
 exit 1
fi

MAKE=`which gmake 2> /dev/null`
if [ -z "$MAKE" ]
then
 echo "Assuming make program is GNU make program"
 MAKE=make
fi

MAKE="$MAKE -j 8"

WGET=`which wget 2> /dev/null`
if [ -z "$WGET" ]
then
 echo "wget programs must be installed to download netCDF lib."
 exit 1
fi

WGET="$WGET --no-check-certificate"

echo "This script installs the netCDF/mpi librares in the"
echo
echo -e "\t $DEST"
echo
echo "directory. If something goes wrong, logs are saved in"
echo
```

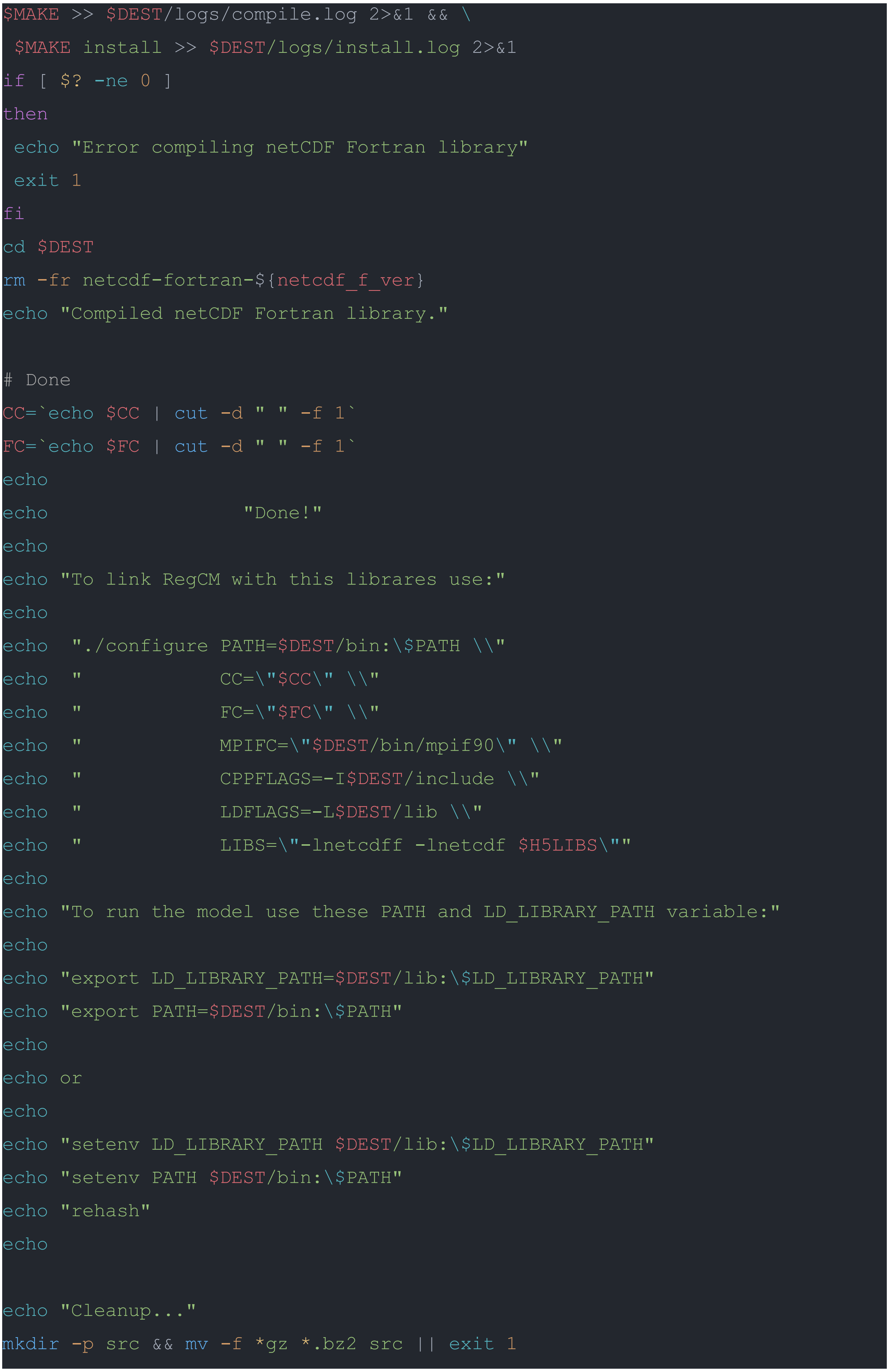
```
echo -e "\t$DEST/logs"
echo

cd $DEST
mkdir -p $DEST/logs
echo "Downloading ZLIB library..."
$WGET -c $ZLIB/zlib-${zlib_ver}.tar.gz -o $DEST/logs/download_Z.log
if [ $? -ne 0 ]
then
 echo "Error downloading ZLIB library from zlib.net"
 exit 1
fi
echo "Downloading HDF5 library..."
$WGET -c $HDFGROUP/hdf5-${hdf5_ver}.tar.gz \
     -o $DEST/logs/download_H.log
if [ $? -ne 0 ]
then
 echo "Error downloading HDF5 library from www.hdfgroup.org"
 exit 1
fi

echo "Downloading PnetCDF Library..."
$WGET -c
https://parallel-netcdf.github.io/Release/pnetcdf-${pnetcdf_ver}.tar.gz -o
$DEST/logs/download_C.log
if [ $? -ne 0 ]
then
 echo "Error downloading PnetCDF library from parallel-netcdf.github.io"
 exit 1
fi

echo "Downloading netCDF Library..."
$WGET -c $UNIDATA/netcdf-c/${netcdf_c_ver}/netcdf-c-${netcdf_c_ver}.tar.gz -o
$DEST/logs/download_C.log
if [ $? -ne 0 ]
then
 echo "Error downloading netCDF C library from www.unidata.ucar.edu"
 exit 1
fi
```

```
$WGET -c
$UNIDATA/netcdf-fortran/${netcdf_f_ver}/netcdf-fortran-${netcdf_f_ver}.tar.gz
-o $DEST/logs/download_F.log
if [ $? -ne 0 ]
then
 echo "Error downloading netCDF Fortran library from www.unidata.ucar.edu"
 exit 1
fi
# echo "Downloading OPENMPI Library..."
# $WGET -c $OPENMPI/openmpi-${ompi_ver}.tar.bz2 -o $DEST/logs/download_M.log
# if [ $? -ne 0 ]
# then
#   echo "Error downloading OPENMPI from OPENMPI website"
#   exit 1
# fi

rm -f logs/*.log

echo "Compiling zlib Library."
tar zxvf zlib-${zlib_ver}.tar.gz >> $DEST/logs/extract.log
if [ $? -ne 0 ]
then
 echo "Error uncompressing zlib library"
 exit 1
fi
cd zlib-${zlib_ver}
echo CC="$CC" FC="$FC" CFLAGS="$CFLAGS" CPPFLAGS="$CPPFLAGS"
FCFLAGS="$FCFLAGS" ./configure --prefix=$DEST --shared >> \
 $DEST/logs/configure.log
CC="$CC" FC="$FC" CFLAGS="$CFLAGS" CPPFLAGS="$CPPFLAGS" FCFLAGS="FCFLAGS"
./configure --prefix=$DEST --shared >> \
            $DEST/logs/configure.log 2>&1
$MAKE >> $DEST/logs/compile.log 2>&1 && \
 $MAKE install >> $DEST/logs/install.log 2>&1
if [ $? -ne 0 ]
then
 echo "Error compiling zlib library"
 exit 1
fi
cd $DEST
rm -fr zlib-${zlib_ver}
```

```
echo "Compiled zlib library."


# echo "Compiling MPI library."
# tar jxvf openmpi-${ompi_ver}.tar.bz2 >> $DEST/logs/extract.log
# if [ $? -ne 0 ]
# then
#   echo "Error uncompressing openmpi library"
#   exit 1
# fi
# cd openmpi-${ompi_ver}
# echo ./configure CC="$CC" FC="$FC" F77="$FC" CXX="$CXX" \
#       --prefix=$DEST --disable-cxx >> $DEST/logs/configure.log
# ./configure CC="$CC" FC="$FC" F77="$FC" CXX="$CXX" \
#   --prefix=$DEST >> $DEST/logs/configure.log 2>&1
# $MAKE >> $DEST/logs/compile.log 2>&1 && \
#   $MAKE install >> $DEST/logs/install.log 2>&1
# if [ $? -ne 0 ]
# then
#   echo "Error compiling openmpi library"
#   exit 1
# fi
# cd $DEST
# rm -fr openmpi-${ompi_ver}
# echo "Compiled MPI library."
echo "Compiling HDF5 library."
tar zxvf hdf5-${hdf5_ver}.tar.gz >> $DEST/logs/extract.log
cd hdf5-${hdf5_ver}
echo ./configure CC="$CC" CXX="$CXX" FC="$FC" CFLAGS="$CFLAGS"
CPPFLAGS="$CPPFLAGS" FCFLAGS="$FCFLAGS" \
       --prefix=$DEST --with-zlib=$DEST --enable-shared \
       --disable-cxx --disable-fortran --enable-parallel --enable-hl >>
$DEST/logs/configure.log
./configure CC="$CC" CXX="$CXX" FC="$FC" CFLAGS="$CFLAGS"
CPPFLAGS="$CPPFLAGS" FCFLAGS="$FCFLAGS" \
 --prefix=$DEST --with-zlib=$DEST --enable-shared \
       --disable-cxx --disable-fortran --enable-parallel --enable-hl >>
$DEST/logs/configure.log 2>&1
$MAKE >> $DEST/logs/compile.log 2>&1 && \
 $MAKE install >> $DEST/logs/install.log 2>&1
if [ $? -ne 0 ]
```

```
then
 echo "Error compiling HDF5 library"
 exit 1
fi
cd $DEST
rm -fr hdf5-${hdf5_ver}
echo "Compiled HDF5 library."

echo "Compiling PnetCDF Library."
tar xf pnetcdf-$pnetcdf_ver.tar.gz >> $DEST/logs/extract.log
cd pnetcdf-$pnetcdf_ver
echo ./configure CC="$CC" CXX="$CXX" FC="$FC" CXXFLAGS="$CPPFLAGS"
CFLAGS="$CFLAGS" \
--with-mpi=/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/openmpi-5.
0.3-ccc6ge624vfgynsmfanradnyfjh6jpnj \
--enable-shared \
--enable-subfiling \
--prefix=$DEST >> $DEST/logs/configure.log

./configure CC="$CC" CXX="$CXX" FC="$FC" CXXFLAGS="$CPPFLAGS"
CFLAGS="$CFLAGS" \
--with-mpi=/opt/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/openmpi-5.
0.3-ccc6ge624vfgynsmfanradnyfjh6jpnj \
--enable-shared \
--enable-subfiling \
--prefix=$DEST >> $DEST/logs/configure.log 2>&1

$MAKE >> $DEST/logs/compile.log 2>&1 && \
 $MAKE install >> $DEST/logs/install.log 2>&1
if [ $? -ne 0 ]
then
 echo "Error compiling PnetCDF library"
 exit 1
fi
cd $DEST
rm -rf pnetcdf-$pnetcdf_ver
echo "Compiled PnetCDF library."

echo "Compiling netCDF Library."
tar zxvf netcdf-c-${netcdf_c_ver}.tar.gz >> $DEST/logs/extract.log
cd netcdf-c-${netcdf_c_ver}
```

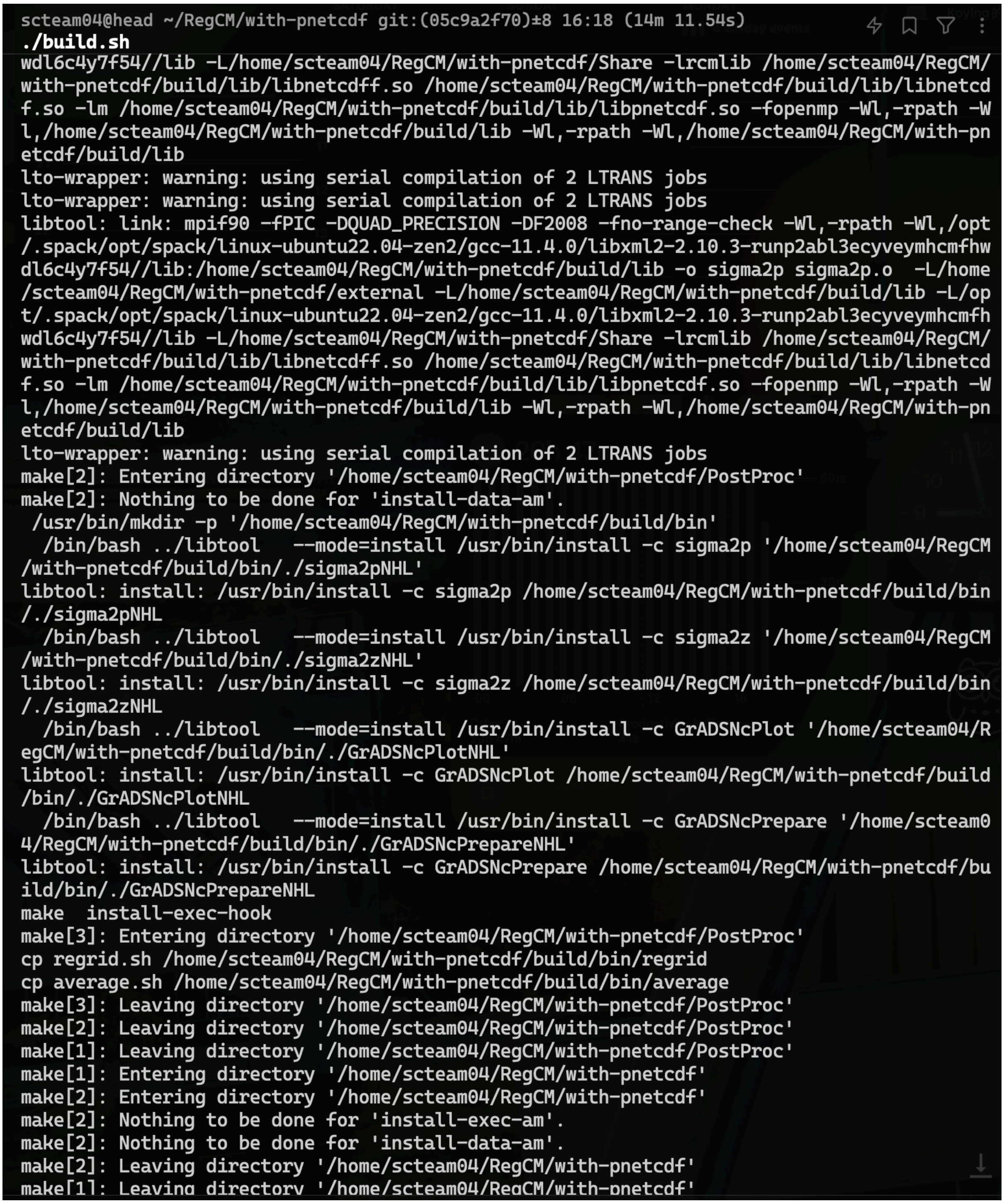
```
H5LIBS="-lhdf5_hl -lhdf5 -lz"
if [ "X$FC" == "Xgfortran" ]
then
 H5LIBS="$H5LIBS -lm -ldl"
fi
echo ./configure CC="$CC" FC="$FC" --prefix=$DEST --enable-netcdf-4 \
 FCFLAGS="$FCFLAGS" CPPFLAGS="$CPPFLAGS -I$DEST/include -I$LIBXML2/include"
LDFLAGS="-L$DEST/lib -L$LIBXML2/lib" LIBS="$H5LIBS" \
 --enable-dap --enable-shared --enable-pnetcdf --enable-mmap >>
$DEST/logs/configure.log
./configure CC="$CC" FC="$FC" --prefix=$DEST --enable-netcdf-4 \
 FCFLAGS="$FCFLAGS" CPPFLAGS="$CPPFLAGS -I$DEST/include -I$LIBXML2/include"
LDFLAGS="-L$DEST/lib -L$LIBXML2/lib" LIBS="$H5LIBS" \
 --enable-dap --enable-shared --enable-pnetcdf --enable-mmap >>
$DEST/logs/configure.log 2>&1
$MAKE >> $DEST/logs/compile.log 2>&1 && \
 $MAKE install >> $DEST/logs/install.log 2>&1
if [ $? -ne 0 ]
then
 echo "Error compiling netCDF C library"
 exit 1
fi


cd $DEST
rm -fr netcdf-c-${netcdf_c_ver}
echo "Compiled netCDF C library."


tar zxvf netcdf-fortran-${netcdf_f_ver}.tar.gz >> $DEST/logs/extract.log
cd netcdf-fortran-${netcdf_f_ver}
echo ./configure PATH=$DEST/bin:$PATH CC="$CC" FC="$FC" F77="$FC" \
    CPPFLAGS="$CPPFLAGS -I$DEST/include" LDFLAGS="-L$DEST/lib"
FCFLAGS="$FCFLAGS" FFLAGS="$FCFLAGS" --prefix=$DEST \
    --enable-shared --enable-parallel-tests --enable-parallel >>
$DEST/logs/configure.log
./configure PATH=$DEST/bin:$PATH CC="$CC" FC="$FC" F77="$FC" \
    CPPFLAGS="$CPPFLAGS -I$DEST/include" LDFLAGS="-L$DEST/lib"
FCFLAGS="$FCFLAGS" FFLAGS="$FCFLAGS" --prefix=$DEST \
    --enable-shared --enable-parallel-tests --enable-parallel >>
$DEST/logs/configure.log 2>&1
```



```
$MAKE >> $DEST/logs/compile.log 2>&1 && \
 $MAKE install >> $DEST/logs/install.log 2>&1
if [ $? -ne 0 ]
then
 echo "Error compiling netCDF Fortran library"
 exit 1
fi
cd $DEST
rm -fr netcdf-fortran-${netcdf_f_ver}
echo "Compiled netCDF Fortran library."

# Done
CC=`echo $CC | cut -d " " -f 1`
FC=`echo $FC | cut -d " " -f 1`
echo
echo                 "Done!"
echo
echo "To link RegCM with this librares use:"
echo
echo  "./configure PATH=$DEST/bin:\$PATH \\"
echo  "            CC=\"$CC\" \\"
echo  "            FC=\"$FC\" \\"
echo  "            MPIFC=\"$DEST/bin/mpif90\" \\"
echo  "            CPPFLAGS=-I$DEST/include \\"
echo  "            LDFLAGS=-L$DEST/lib \\"
echo  "            LIBS=\"-lnetcdff -lnetcdf $H5LIBS\""
echo
echo "To run the model use these PATH and LD_LIBRARY_PATH variable:"
echo
echo "export LD_LIBRARY_PATH=$DEST/lib:\$LD_LIBRARY_PATH"
echo "export PATH=$DEST/bin:\$PATH"
echo
echo or
echo
echo "setenv LD_LIBRARY_PATH $DEST/lib:\$LD_LIBRARY_PATH"
echo "setenv PATH $DEST/bin:\$PATH"
echo "rehash"
echo

echo "Cleanup..."
mkdir -p src && mv -f *gz *.bz2 src || exit 1
```

```
exit 0
```

2. build screenshot



```
scteam04@head ~/RegCM/with-pnetcdf git:(05c9a2f70)±8 16:18 (14m 11.54s)
./build.sh
wdl6c4y7f54//lib -L/home/scteam04/RegCM/with-pnetcdf/Share -lrcmlib /home/scteam04/RegCM/
with-pnetcdf/build/lib/libnetcdff.so /home/scteam04/RegCM/with-pnetcdf/build/lib/libnetcd
f.so -lm /home/scteam04/RegCM/with-pnetcdf/build/lib/libpnetcdf.so -fopenmp -Wl,-rpath -W
l,/home/scteam04/RegCM/with-pnetcdf/build/lib -Wl,-rpath -Wl,/home/scteam04/RegCM/with-pn
etcdf/build/lib
lto-wrapper: warning: using serial compilation of 2 LTRANS jobs
lto-wrapper: warning: using serial compilation of 2 LTRANS jobs
libtool: link: mpif90 -fPIC -DQUAD_PRECISION -DF2008 -fno-range-check -Wl,-rpath -Wl,/opt
/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/libxml2-2.10.3-runp2abl3ecyveymhcmfhw
dl6c4y7f54//lib:/home/scteam04/RegCM/with-pnetcdf/build/lib -o sigma2p sigma2p.o  -L/home
/scteam04/RegCM/with-pnetcdf/external -L/home/scteam04/RegCM/with-pnetcdf/build/lib -L/op
t/.spack/opt/spack/linux-ubuntu22.04-zen2/gcc-11.4.0/libxml2-2.10.3-runp2abl3ecyveymhcmfh
wdl6c4y7f54//lib -L/home/scteam04/RegCM/with-pnetcdf/Share -lrcmlib /home/scteam04/RegCM/
with-pnetcdf/build/lib/libnetcdff.so /home/scteam04/RegCM/with-pnetcdf/build/lib/libnetcd
f.so -lm /home/scteam04/RegCM/with-pnetcdf/build/lib/libpnetcdf.so -fopenmp -Wl,-rpath -W
l,/home/scteam04/RegCM/with-pnetcdf/build/lib -Wl,-rpath -Wl,/home/scteam04/RegCM/with-pn
etcdf/build/lib
lto-wrapper: warning: using serial compilation of 2 LTRANS jobs
make[2]: Entering directory '/home/scteam04/RegCM/with-pnetcdf/PostProc'
make[2]: Nothing to be done for 'install-data-am'.
 /usr/bin/mkdir -p '/home/scteam04/RegCM/with-pnetcdf/build/bin'
  /bin/bash ../libtool   --mode=install /usr/bin/install -c sigma2p '/home/scteam04/RegCM
/with-pnetcdf/build/bin/./sigma2pNHL'
libtool: install: /usr/bin/install -c sigma2p /home/scteam04/RegCM/with-pnetcdf/build/bin
/./sigma2pNHL
  /bin/bash ../libtool   --mode=install /usr/bin/install -c sigma2z '/home/scteam04/RegCM
/with-pnetcdf/build/bin/./sigma2zNHL'
libtool: install: /usr/bin/install -c sigma2z /home/scteam04/RegCM/with-pnetcdf/build/bin
/./sigma2zNHL
  /bin/bash ../libtool   --mode=install /usr/bin/install -c GrADSNcPlot '/home/scteam04/R
egCM/with-pnetcdf/build/bin/./GrADSNcPlotNHL'
libtool: install: /usr/bin/install -c GrADSNcPlot /home/scteam04/RegCM/with-pnetcdf/build
/bin/./GrADSNcPlotNHL
  /bin/bash ../libtool   --mode=install /usr/bin/install -c GrADSNcPrepare '/home/scteam0
4/RegCM/with-pnetcdf/build/bin/./GrADSNcPrepareNHL'
libtool: install: /usr/bin/install -c GrADSNcPrepare /home/scteam04/RegCM/with-pnetcdf/bu
ild/bin/./GrADSNcPrepareNHL
make  install-exec-hook
make[3]: Entering directory '/home/scteam04/RegCM/with-pnetcdf/PostProc'
cp regrid.sh /home/scteam04/RegCM/with-pnetcdf/build/bin/regrid
cp average.sh /home/scteam04/RegCM/with-pnetcdf/build/bin/average
make[3]: Leaving directory '/home/scteam04/RegCM/with-pnetcdf/PostProc'
make[2]: Leaving directory '/home/scteam04/RegCM/with-pnetcdf/PostProc'
make[1]: Leaving directory '/home/scteam04/RegCM/with-pnetcdf/PostProc'
make[1]: Entering directory '/home/scteam04/RegCM/with-pnetcdf'
make[2]: Entering directory '/home/scteam04/RegCM/with-pnetcdf'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/scteam04/RegCM/with-pnetcdf'
make[1]: Leaving directory '/home/scteam04/RegCM/with-pnetcdf'
```