Compile GFR

Jingchang Shi

2019-04-16

---

1 Compile Contrib

The original instructions from GFR say that we need to download and compile the following packages.

- metis-5.1.0.tar.gz, link
- mpich-3.2.tar.gz, link
- hdf5-1.8.19.tar.gz, link
- CGNS-3.3.0.tar.gz, link

Here we compile only 3 of them, excluding mpich. Supercomputers like Tianhe-2 have their customized MPI environments. Using our own MPI will degrade the running speed.

1.1 File Structure

```
.
├── contrib
│   ├── cgns
│   ├── hdf5
│   └── metis
├── contrib_src
│   ├── cgns_src
│   ├── hdf5_src
│   └── metis_src
└── GFR
    ├── build
    └── src
```

Before compiling the contrib packages, set the root dir.

```
export gfrdir=/home/jcshi/Documents/gfr_project
```

Also check your compilation environments.

```
jcshi@BigMachine:~/Documents/gfr_project$ which gcc
/usr/bin/gcc
jcshi@BigMachine:~/Documents/gfr_project$ gcc --version
gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

jcshi@BigMachine:~/Documents/gfr_project$ which mpicc
/usr/bin/mpicc
jcshi@BigMachine:~/Documents/gfr_project$ which mpif90
/usr/bin/mpif90
jcshi@BigMachine:~/Documents/gfr_project$ mpichversion
MPICH Version:      3.3a2
MPICH Release date: Sun Nov 13 09:12:11 MST 2016
MPICH Device:       ch3:nemesis
MPICH configure:    --build=x86_64-linux-gnu --prefix=/usr --
includedir=${prefix}/include --mandir=${prefix}/share/man --
infodir=${prefix}/share/info --sysconfdir=/etc --localstatedir=/var --disable-silent-
rules --libdir=${prefix}/lib/x86_64-linux-gnu --libexecdir=${prefix}/lib/x86_64-linux-
gnu --disable-maintainer-mode --disable-dependency-tracking --with-libfabric --enable-
shared --prefix=/usr --enable-fortran=all --disable-rpath --disable-wrapper-rpath --
sysconfdir=/etc/mpich --libdir=/usr/lib/x86_64-linux-gnu --
includedir=/usr/include/mpich --docdir=/usr/share/doc/mpich --with-hwloc-prefix=system
--enable-checkpointing --with-hydra-ckpointlib=blcr CPPFLAGS= CFLAGS= CXXFLAGS=
FFLAGS= FCFLAGS=
MPICH CC:   gcc  -g -O2 -fdebug-prefix-map=/build/mpich-O9at2o/mpich-3.3~a2=. -fstack-
protector-strong -Wformat -Werror=format-security  -O2
MPICH CXX:  g++  -g -O2 -fdebug-prefix-map=/build/mpich-O9at2o/mpich-3.3~a2=. -fstack-
protector-strong -Wformat -Werror=format-security -O2
MPICH F77:  gfortran  -g -O2 -fdebug-prefix-map=/build/mpich-O9at2o/mpich-3.3~a2=. -
fstack-protector-strong -O2
MPICH FC:   gfortran  -g -O2 -fdebug-prefix-map=/build/mpich-O9at2o/mpich-3.3~a2=. -
fstack-protector-strong -O2
MPICH Custom Information:
```

1.2 Metis

```
cd ${gfrdir}/contrib_src
wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/metis-5.1.0.tar.gz
tar -xzvf metis-5.1.0.tar.gz
cd metis-5.1.0
make cc=gcc prefix=${gfrdir}/contrib/metis config
make
make install
cd ${gfrdir}
```

1.3 HDF5

```
cd ${gfrdir}/contrib_src
wget https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8/hdf5-1.8.19/src/hdf5-
1.8.19.tar.gz
tar -xzvf hdf5-1.8.19.tar.gz
cd hdf5-1.8.19
mkdir -p build
cd build
cmake -DHDF5_BUILD_CPP_LIB:BOOL=OFF -DHDF5_BUILD_FORTRAN:BOOL=ON -
DHDF5_ENABLE_PARALLEL:BOOL=ON -DCMAKE_INSTALL_PREFIX:STRING=${gfrdir}/contrib/hdf5 -
DCMAKE_BUILD_TYPE:STRING=Release -DBUILD_TESTING:BOOL=OFF -
DHDF5_BUILD_EXAMPLES:BOOL=OFF  ..
make
```

```
make install
cd ${gfrdir}
```

Here, the original instructions from GFR let you use the `configure` way of compiling applications. HDF5 officially offer 2 ways: `configure` way and `cmake` way. The `configure` way will not put the cmake files to the expected location thus cause an error of HDF5 package not found when using cmake configuring CGNS. So, we use the `cmake` way.

1.4 CGNS

```
cd ${gfrdir}/contrib_src
wget https://github.com/CGNS/CGNS/archive/v3.3.0.tar.gz
mv v3.3.0.tar.gz CGNS-3.3.0.tar.gz
tar -xzvf CGNS-3.3.0.tar.gz
cd CGNS-3.3.0
mkdir -p build
cd build
export HDF5_DIR=${gfrdir}/contrib/hdf5/share/cmake
cmake -D CGNS_ENABLE_FORTRAN:BOOL=ON -D CGNS_ENABLE_HDF5:BOOL=ON -D
HDF5_NEED_MPI:BOOL=ON -D CGNS_ENABLE_PARALLEL:BOOL=ON -D CMAKE_C_COMPILER:STRING=mpicc
-D CMAKE_Fortran_COMPILER:STRING=mpif90 -D
CMAKE_INSTALL_PREFIX:PATH=${gfrdir}/contrib/cgns ..
cd src/tools/CMakeFiles/cgnscheck.dir && awk '{$NF=$NF" -lhdf5"; print}' link.txt >
link && mv link link.txt && cd -
cd src/tools/CMakeFiles/cgnscompress.dir && awk '{$NF=$NF" -lhdf5"; print}' link.txt >
link && mv link link.txt && cd -
cd src/tools/CMakeFiles/cgnsconvert.dir && awk '{$NF=$NF" -lhdf5"; print}' link.txt >
link && mv link link.txt && cd -
cd src/tools/CMakeFiles/cgnsdiff.dir && awk '{$NF=$NF" -lhdf5"; print}' link.txt >
link && mv link link.txt && cd -
cd src/tools/CMakeFiles/cgnslist.dir && awk '{$NF=$NF" -lhdf5"; print}' link.txt >
link && mv link link.txt && cd -
export LIBRARY_PATH=${gfrdir}/contrib/hdf5/lib
make
make install
cd ${gfrdir}
```

In the above shell operations, the files `link.txt` are appended with `-lhdf5` to resolve the issue of `unreferenced link to H5 functions` when linking those CGNS tools. In fact, at that stage, the static and shared library of CGNS are already compiled successfully.

Also, the linking process requires HDF5 lib directory to be known to CGNS. Thus we `export LIBRARY_PATH`. Not sure why libhdf5 is still not found when CMake already finds where HDF5 locates.

2 Compile GFR

The environments are set up in the `*.mk` files.

```
cd ${gfrdir}/GFR/build
make
```

Now, `debug_gfr` and `gfr` will be put in `${gfrdir}/GFR/build/bin`.

3 Run GFR

Each time running GFR, you have to set up the lib search path.

```
export gfrdir=/home/jcshi/Documents/gfr_project
export LD_LIBRARY_PATH=${gfrdir}/contrib/metis/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=${gfrdir}/contrib/hdf5/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=${gfrdir}/contrib/cgns/lib:$LD_LIBRARY_PATH
```

Then you can run GFR in the serial mode or parallel mode.