Parallel Tools User Guide

Introduced with

HDF5 Release 1.13.0

December 6, 2021

The HDF Group

http://www.HDFGroup.org

Copyright Notice and License Terms

See the Copyright Notices page on the HDF Group web site for the HDF5 Copyright Notice and Licensing Terms. This information can also be found in COPYING file in the top directory of the HDF5 source code.

Documentation

See the HDF Support Portal for documentation a information on getting help.

1. Introduction

This document introduces parallel tools for HDF5. The initial development implements a new tool based on a set of 3rd party open-source libraries collectively known as mpiFileUtils. This approach can greatly enhance the serial hdf5 tool performance over large collections of files by utilizing MPI parallelism to distribute an application load over many independent MPI ranks and files. The current serial tool functionality is retained and even enhanced in some areas; particularly by adding a capability to capture tool outputs in text or in the future as HDF5 formatted files. The purpose of this document is introduce the new parallel (h5dwalk) tool and to provide details of how to build and run simple parallel examples.

HDF5 tools are principally informational, e.g. `h5dump` and `h5ls` are "viewers" which allow users to examine the contents of an existing HDF5 file. The `h5diff` command for example, is used to compare files and present the differences if any, in a human readable text format. Eventually, it should be a goal to expand on the available format(s) by which tool outputs can be recorded. Typically, the output will be generated by applying output filters on the tool output stream. The newest tool `h5dwalk`, is discussed in more detail in the following section. It provides parallelism for improved performance while also including critical logging capabilities to capture outputs from applying the serial tools over large file collections.

2. The h5dwalk utility

The `h5dwalk` utility provides a parallel alternative to creating and running script based approaches to invoke serial HDF5 tools on a collection of hdf5 files. As a means of invoking parallel instances of a serial tool, the `h5dwalk` application can accept directories as input arguments. This new tool provides recursive file discovery and filtering to select hdf5 formatted files. The resulting file collection is distributed between MPI ranks and individual files are then selected for input to a user selected application. Figure below, shows the current runtime options for `h5dwalk`.

```
[bin]$ ./h5dwalk --help

Usage: ./h5dwalk [options] <path> ...

Options:
  -i, --input <file>      - read list from file
  -o, --output <file>     - write output summary to the named file.
  -E, --error  <file>     - write processed errors to file in text format
  -l, --log_text <dir>    - write individual tool outputs to a file. Logs can be written to an optional
named directory.
  -T, --tool <executable> - name of the HDF5 tool to invoke
  -h, --help              - print usage

For more information see https://mpifileutils.readthedocs.io.

[bin]$
```

Figure 1: h5dwalk runtime options

As mentioned previously, the HDF5 tools collection serves to view or to possibly modify the contents of an existing HDF5 formatted file. Users can for example, discover the number and naming of groups, datasets, and attributes contained within a file by utilizing `h5ls` or `h5dump`.
Figure 2 below, shows an example of running 'h5dump -n' on a collection of 376 HDF5 files located in a directory ("`/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/testfiles`") with all output directed to the named logfile ("show-h5dump-h5files.log").

```
[ bin]$ mpiexec -n 4 ./h5dwalk -o show-h5dump-h5files.log -T ./h5dump
$HOME/Sandbox/HDF5/GITHUB/hdf5/tools/testfiles
[ bin]$ more show-h5dump-h5files.log

---------
Command: ./h5dump -n /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/testfiles/tnestedcmpddt.h5
HDF5 "/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/testfiles/tnestedcmpddt.h5" {
FILE_CONTENTS {
 group      /
 dataset    /dset1
 dataset    /dset2
 dataset    /dset4
 dataset    /dset5
 datatype   /enumtype
 group      /group1
 dataset    /group1/dset3
 datatype   /type1
 }
}
...
```

Figure 2: h5dwalk example

The log files show each hdf5 tool output instance, prefixed by the actual command line used to invoke the tool. When selecting logfile generation using `-l (--log_text )`, each independent tool instance will have an associated logfile whose file name is a combination of the 1st hdf5 file in the tool argument

list, with the actual tool-name which generated the logfile text. For h5dwalk examples which require multiple hdf5 files, e.g. for h5diff (which compares two hdf5 files), there can be an file ordering issue due to the way directory traversals are implemented. The ideal implementation should match file_N from directory_1 and pair that with file_N from directory 2. This "ideal" is not actually implemented nor desired in many cases, i.e. even when contents of directory_1 and directory_2 are identical, the parallel tree walking algorithm may provide randomness. In other instances, file matching might be more advantageous when all files are from a single directory. In this latter instance, we don't have a fixed algorithm to select a perfect "pairing" for all cases. There are two supported approaches which give users complete control over file pairing:

1. The h5dwalk implementation supports @filename indirections, where "filename" contains a list of hdf5 filenames to be used in the order specified by their position (one filename-per-line). For the h5diff tool case, file#1 from the 1st indirect file will be paired with file#1 from the 2nd indirect file.
2. H5dwalk also supports a `–input <filename>` option which basically allows a script approach to be used in place of the indirect file or directory traversals.

The file indirection approach provides an easy specification of file matching but only allows a single set of tool runtime arguments (those provided on the command line with h5dwalk).

```
[riwarren@rawlinux bin]$ more ../../tools/srcfiles.txt
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_strings1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_eps1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr_v_level1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_dset1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_hyper1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_dset_zero_dim_size1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/non_comparables1.h5
[riwarren@rawlinux bin]$
[riwarren@rawlinux bin]$ more ../../tools/destfiles.txt
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_strings2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_eps2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr_v_level2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_dset2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_hyper2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_dset_zero_dim_size2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/non_comparables2.h5
[riwarren@rawlinux bin]$
[riwarren@rawlinux bin]$
[riwarren@rawlinux bin]$ mpiexec -n 2 ./h5dwalk -o show_indirect_files.log -T ./h5diff \
@../../tools/srcfiles.txt @../../tools/destfiles.txt
```

Figure 3: h5dwalk example using two indirect files to specify inputs for h5diff

In figure 3, `h5dwalk` invokes the `h5diff` tool with indirect files whose contents are shown. The approach facilitates the use case where all selected files are contained within the same file system directory. In this example, we can notice from the output log (`show_indirect_files.log`), that the tool selects files from each indirect file list by their shared index, i.e. file_1 from srcfiles.txt will be paired with file_1 from destfiles.txt and passed as input arguments to h5diff.

```
[ bin]$ cat show_indirect_files.log

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5
dataset: </g1/dset1> and </g1/dset1>
5 differences found

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_eps1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_eps2.h5
dataset: </DS1> and </DS1>
28 differences found

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr_v_level1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr_v_level2.h5
attribute: <integer1 of </dset>> and <integer1 of </dset>>
2 differences found
attribute: <float1 of </g>> and <float1 of </g>>
2 differences found
attribute: <integer1 of </g>> and <integer1 of </g>>
2 differences found
attribute: <float2 of </g2>> and <float2 of </g2>>
2 differences found
attribute: <integer1 of </g2>> and <integer1 of </g2>>
2 differences found

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_hyper1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_hyper2.h5
dataset: </big> and </big>
1024 differences found

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/non_comparables1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/non_comparables2.h5
attribute: <attr of </g1/dset1>> and <attr of </g1/dset1>>
3 differences found
dataset: </g1/dset2> and </g1/dset2>
3 differences found
dataset: </g2/dset1> and </g2/dset1>
3 differences found
attribute: <attr4 of </g2/dset1>> and <attr4 of </g2/dset1>>
3 differences found
dataset: </g2/dset2> and </g2/dset2>
3 differences found
--------------------------------
Some objects are not comparable
--------------------------------
Use -c for a list of objects.

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_strings1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_strings2.h5
dataset: </string1> and </string1>
4 differences found
dataset: </string2> and </string2>
24 differences found
dataset: </string3> and </string3>
31 differences found
dataset: </string4> and </string4>
4 differences found

---------
Command: ./h5diff /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_attr2.h5
attribute: <VLstring of </>> and <VLstring of </>>
4 differences found
attribute: <VLstring2D of </>> and <VLstring2D of </>>
12 differences found
attribute: <VLstring3D of </>> and <VLstring3D of </>>
47 differences found
attribute: <array of </>> and <array of </>>
6 differences found
attribute: <array2D of </>> and <array2D of </>>
18 differences found
attribute: <array3D of </>> and <array3D of </>>
72 differences found
```

Figure 4: The contents of the "show_indirect_files.log" generated by h5diff

The scripting approach allows virtually any combination of tools, files, and tool arguments, but improves upon a simple scripting approach by load balancing the execution across the MPI ranks.

```
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5 g1/dset1 g1/dset2
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff -r \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff -r \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5 g1/dset1 g1/dset2
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff --report --delta=5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5 g1/dset3 g1/dset4
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff -v -p 0.02 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 g1/dset5 g1/dset6
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff --verbose --relative=0.02 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 \
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5 g1/dset7 g1/dset8
```

Figure 5: First few lines of a sample script file (demo-h5dwalk.txt)

```
[ bin]$ mpiexec -n 4 ./h5dwalk -i ../../tools/test/demo-h5dwalk.txt -o showme-demo.log
```

The example shown above is run using 4 cores and performs nearly twice as quickly as running on 2 cores and generates a log file with the name “showme-demo.log”. Figure 6 (below) provides a look at the first lines in the resulting logfile. It shows that for this 4 MPI rank example, that every 4th script line is shown. This has to do with the way the script is distributed between MPI ranks and eventually printed, i.e. the script line-number modulo 4 (total number of MPI ranks) will match the MPI RANK of the process executing the script line.

Figure 6: Portion of the "showme-demo.log"

```
---------
Command: /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff-shared
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5
dataset: </g1/dset1> and </g1/dset1>
5 differences found

---------
Command: /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff-shared --report --delta=5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5 g1/dset3 g1/dset4
dataset: </g1/dset3> and </g1/dset4>
size:           [3x2]           [3x2]
position        dset3           dset4           difference
----------------------------------------------------------
[ 0 1 ]          100             120             20
[ 1 0 ]          100             160             60
[ 2 0 ]          100             80              20
[ 2 1 ]          100             40              60
4 differences found

---------
Command: /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin/h5diff-shared -v
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic1.h5
/home/riwarren/Sandbox/HDF5/GITHUB/hdf5/tools/test/h5diff/testfiles/h5diff_basic2.h5

file1     file2
---------------------------------------
    x      x    /
    x      x    /g1
    x           /g1/d1
    x           /g1/d2
    x      x    /g1/dset1
    x           /g1/dset10
    x           /g1/dset11
    x           /g1/dset12
           x    /g1/dset2
    x           /g1/dset3
           x    /g1/dset4
    x           /g1/dset5
    x           /g1/dset6
    x           /g1/dset7
    x           /g1/dset8
    x           /g1/dset9
    x           /g1/fp1
    x           /g1/fp15
    x           /g1/fp16
    x           /g1/fp17
    x           /g1/fp18
    x           /g1/fp18_COPY
    x           /g1/fp19
    x           /g1/fp19_COPY
    x           /g1/fp2
    x           /g1/fp20
    x           /g1/fp20_COPY
    x           /g1/ld
           x    /g2
           x    /g2/dset1
           x    /g2/dset2
           x    /g2/dset3
           x    /g2/dset4
           x    /g2/dset5
           x    /g2/dset6
           x    /g2/dset7
           x    /g2/dset8
           x    /g2/dset9

group  : </> and </>
0 differences found
group  : </g1> and </g1>
0 differences found
dataset: </g1/dset1> and </g1/dset1>
size:           [3x2]           [3x2]
position        dset1           dset1           difference
----------------------------------------------------------
[ 0 0 ]          1               0               1
[ 0 1 ]          1               1.1             0.1
[ 1 0 ]          1               1.01            0.01
[ 1 1 ]          1               1.001           0.001
[ 2 1 ]          0               1               1
5 differences found
```

2.1. Testing

Parallel testing of `h5dwalk` with other HDF5 tools may require the setting of the LD_LIBRARY_PATH to enable the loader to locate the MPI libraries and binaries as well as the libmfu components. The test scripts shown in the previous examples are provided in the tools/test directory for the hdf5 distribution.

2.2. Building h5dwalk

While h5dwalk is integrated into the HDF5 toolset build, the software includes 3rd party external open-source software dependencies. These external libraries are not included in the HDF5 source code distribution. To enable this functionality, download and build the software found at the mpiFileUtils web site. Once these software dependencies are built and installed, an HDF5 library and tools build can proceed.

2.2.1. Autotools

For users of `autotools`, the starting point for initiating an HDF5 build is to run the configure script. Users have several build options ranging from choosing a 'debug' or 'release' build, to choosing library extensions such as compression libraries or in our case, to incorporate mpiFileUtils into the build process. The actual build of h5dwalk requires two configuration switches, i.e.

- Select a parallel library build (--enable-parallel) ; and
- Enable the use of libmfu (--with-libmfu)

```
[ hdf5]$ ./configure --enable-parallel --enable-build-mode=debug --prefix=$HOME –enable-parallel-tools
--with-libmfu=$HOME
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
```

Figure 7: Example configure script execution

In the example `–with-libmfu=$HOME`, we indicate that libmfu components are installed in subdirectories of $HOME, i.e. $HOME/include and $HOME/lib. Once the configure script is run and all makefiles have been generated, the user should be able to simply invoke the 'make' command to build the library and tools.

2.2.2. CMake

For user of CMake, the build process achieves a result similar to that described in the Autotools section. We enable a parallel library and parallel_tools build flags. Before running ccmake, the user should provide a CMAKE hint to help locate the libmfu software. This is accomplished by setting and environment variable, e.g. "`export MFU_ROOT=$HOME`".
Once the user config selections are defined, the user can type 'c' to configure their selections. This process can be repeated until the 'g' option is enabled. Typing 'g' should generate the necessary Makefile files and then exit.
Upon exit from the cmake selection tool, the user should be able type the 'make' command and if everything has been specified currently, the build process should generate an HDF5 library and the complete set of HDF5 tools.

```
                                                Page 1 of 2
BUILD_SHARED_LIBS                ON
BUILD_STATIC_EXECS               OFF
BUILD_STATIC_LIBS                ON
BUILD_TESTING                    ON
BUILD_USER_DEFINED_LIBS          OFF
CMAKE_ARCHIVE_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_BUILD_TYPE                 Debug
CMAKE_Fortran_MODULE_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_INSTALL_PREFIX             /usr/local/HDF_Group/HDF5/1.13.0
CMAKE_LIBRARY_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_RUNTIME_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CTEST_TEST_TIMEOUT               1200
DEFAULT_API_VERSION              v114
ENABLE_EXTENDED_TESTS            OFF
FETCHCONTENT_BASE_DIR            /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/_deps
FETCHCONTENT_FULLY_DISCONNECTE   OFF
FETCHCONTENT_QUIET               ON
FETCHCONTENT_UPDATES_DISCONNEC   OFF
HDF5_ALLOW_EXTERNAL_SUPPORT      NO
HDF5_BATCH_H5DETECT              OFF
HDF5_BUILD_CPP_LIB               OFF
HDF5_BUILD_DOC                   OFF
HDF5_BUILD_EXAMPLES              ON
HDF5_BUILD_FORTRAN               OFF
HDF5_BUILD_GENERATORS            OFF
HDF5_BUILD_HL_LIB                ON
HDF5_BUILD_JAVA                  OFF
HDF5_BUILD_PARALLEL_TOOLS        ON
HDF5_BUILD_TOOLS                 ON
HDF5_BUILD_UTILS                 ON
HDF5_DISABLE_COMPILER_WARNINGS   OFF
HDF5_ENABLE_ALL_WARNINGS         ON
HDF5_ENABLE_COVERAGE             OFF

BUILD_SHARED_LIBS: Build Shared Libraries
Press [enter] to edit option Press [d] to delete an entry                CMake Version 3.14.5
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

```
                                                Page 1 of 2
BUILD_SHARED_LIBS                ON
BUILD_STATIC_EXECS               OFF
BUILD_STATIC_LIBS                ON
BUILD_TESTING                    ON
BUILD_USER_DEFINED_LIBS          OFF
CMAKE_ARCHIVE_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_BUILD_TYPE                 Debug
CMAKE_Fortran_MODULE_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_INSTALL_PREFIX             /usr/local/HDF_Group/HDF5/1.13.0
CMAKE_LIBRARY_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CMAKE_RUNTIME_OUTPUT_DIRECTORY   /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/bin
CTEST_TEST_TIMEOUT               1200
DEFAULT_API_VERSION              v114
ENABLE_EXTENDED_TESTS            OFF
FETCHCONTENT_BASE_DIR            /home/riwarren/Sandbox/HDF5/GITHUB/hdf5/build/_deps
FETCHCONTENT_FULLY_DISCONNECTE   OFF
FETCHCONTENT_QUIET               ON
FETCHCONTENT_UPDATES_DISCONNEC   OFF
HDF5_ALLOW_EXTERNAL_SUPPORT      NO
HDF5_BATCH_H5DETECT              OFF
HDF5_BUILD_CPP_LIB               OFF
HDF5_BUILD_DOC                   OFF
HDF5_BUILD_EXAMPLES              ON
HDF5_BUILD_FORTRAN               OFF
HDF5_BUILD_GENERATORS            OFF
HDF5_BUILD_HL_LIB                ON
HDF5_BUILD_JAVA                  OFF
HDF5_BUILD_PARALLEL_TOOLS        ON
HDF5_BUILD_TOOLS                 ON
HDF5_BUILD_UTILS                 ON
HDF5_DISABLE_COMPILER_WARNINGS   OFF
HDF5_ENABLE_ALL_WARNINGS         ON
HDF5_ENABLE_COVERAGE             OFF

BUILD_SHARED_LIBS: Build Shared Libraries
Press [enter] to edit option Press [d] to delete an entry                CMake Version 3.14.5
Press [c] to configure       Press [g] to generate and exit
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```