HDF5 Compression Product Quick-start Guide

(Linux/MacOS)

Frank Willmore
Frank.Willmore@hdfgroup.org

HDF5 supports compression of data using a stackable pipeline of filters which can be implemented for reading and writing datasets, both at runtime and post-process. These filters are supported as dynamically loadable plugins, and users can even implement custom filters of their own design.

Requirements:

- The compression library assumes that HDF5 is installed to the directory pointed to by the environment variable HDF5_HOME, e.g. HDF5_HOME=/usr/local/hdf5. It can be installed in user space or admin space, as long as this variable is defined properly.
- The installed version of HDF5 must have been built with `--enable-static=no` in order to force the hdf5 tools to use dynamic linking, which is required by the plugin.
- The compression library also assumes that libz and libsz are installed and will complain if it cannot find these.

Installation:

- Download the plugin from (<<<URL>>>)
- Unpack the plugin archive (will create directory hdf):
  ```
  ~$ tar -zxvf HDF5PluginsCMake-1.10-centos7_64.tar.gz
  ~$ cd hdf
  ~$ ./h5pl-1.0.0-Linux.sh
  -or-
  ~$ ./h5pl-1.0.0-Darwin.sh
  ```
- This self-extracting archive will create directories:
  ```
  ~/hdf/h5pl-1.0.0-Darwin/HDF_Group/HDF5/1.10.2/lib/plugin
  -or-
  ~/hdf/h5pl-1.0.0-Linux/HDF_Group/HDF5/1.10.2/lib/plugin
  ```
- This directory contains shared library (.so) plugin files, one for each compression filter.
- By default, HDF5 looks for plugins to be installed in $HDF5_HOME/lib/plugin. To complete installation:
  ```
  $cp -R ~/hdf/h5pl-1.0.0-Linux/HDF_Group/HDF5/1.10.2/lib/plugin $HDF5_HOME/lib
  -or-
  $cp -R ~/hdf/h5pl-1.0.0-Darwin/HDF_Group/HDF5/1.10.2/lib/plugin $HDF5_HOME/lib
  ```
- If installing to a different location, or to test without installing, make sure that the environment variable HDF5_PLUGIN_PATH is defined to include the location of the plugins (minus the directory ‘plugin’). For example, to use the plugins from their unpack location, simply define:
  ```
  $ HDF5_PLUGIN_PATH=~/hdf/h5pl-1.0.0-Darwin/HDF_Group/HDF5/1.10.2/lib/plugin
  -or-
  $ HDF5_PLUGIN_PATH=~/hdf/h5pl-1.0.0-Linux/HDF_Group/HDF5/1.10.2/lib/plugin
  ```

Testing with HDF5 tools:

If the hdf5 tools are built with dynamic linking, then they can be used to compress/decompress individual datasets or entire files. For example, h5repack can read the data compressed using BLOSC filter (filter info is in the file and need not be included on command line) and repack it using the SZIP (registered as #307) filter:

```
$ h5repack -f UD=307,1,9 h5ex_d_blosc.h5 repacked.h5
```

H5dump is able to read and display compressed data, and will list any filters if including the -p option:

```
$ h5dump -pH repacked.h5
HDF5 "repacked.h5" {
GROUP "/" {
   DATASET "DS1" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE  SIMPLE { ( 32, 64 ) / ( 32, 64 ) }
      STORAGE_LAYOUT {
         CHUNKED ( 4, 8 )
         SIZE 6410 (1.278:1 COMPRESSION)
      }
      FILTERS {
         USER_DEFINED_FILTER {
            FILTER_ID 307
            COMMENT HDF5 bzip2 filter; see
http://www.hdfgroup.org/services/contributions.html
            PARAMS { 9 }
         }
      }
      FILLVALUE {
         FILL_TIME H5D_FILL_TIME_IFSET
         VALUE  H5D_FILL_VALUE_DEFAULT
      }
      ALLOCATION_TIME {
         H5D_ALLOC_TIME_INCR
      }
   }
}
}
```

Using Filters:

Apply filters on the dataset creation property list:

```
herr_t status = H5Pset_filter (dcpl_id, H5Z_FILTER_ZFP, H5Z_FLAG_OPTIONAL, 0, NULL);
```

To check availability of a filter:

```
htri_t avail = H5Zfilter_avail(H5Z_FILTER_ZFP);
```

When reading a dataset, request filter information (if any) from the creation property list:

```
hid_t dcpl_id = H5Dget_create_plist (dset_id);
H5Z_filter_t filter_id = H5Pget_filter2 (dcpl_id, (unsigned) 0, &flags, &nelmts,
values_out, sizeof(filter_name), filter_name, NULL);
```

Examples:

The archive unpacked above also contains example codes in the directory:
`~/hdf/h5pl-1.0.0-Darwin/HDF_Group/HDF5/1.10.2/share/HDFPLExamples/example`
-or-
`~/hdf/h5pl-1.0.0-Linux/HDF_Group/HDF5/1.10.2/share/HDFPLExamples/example`
To build the examples, compile using the familiar h5cc wrapper and run:

```
$ for i in h5ex_d_*.c
$ do
$     h5cc $i
$     ./a.out
$ done
```

These examples are all very similar and show how each of the filters is used.

Filter example/test parameters:

BLOSC	BZIP2	LZ4	LZF	MAFISC	H5Z-ZFP
UD=32001,0,0	UD=307,0,1,9	UD=32004,0,1,3	UD=32000,0,3,0,0,0	UD=32002,0,1,0	UD=32013,1,0,0

Additional reading:

https://support.hdfgroup.org/HDF5/doc/Advanced/DynamicallyLoadedFilters/HDF5DynamicallyLoadedFilters.pdf
https://support.hdfgroup.org/HDF5/doc/H5.user/Filters.html