1\_Primer

Audience: A general user of the HDF5 library

An HDF5 file consists of two kinds of file space. The first kind is metadata, which the HDF5 library uses to describe itself and to identify objects like groups, attributes, datasets etc. Examples of metadata are superblock, object header, B-tree, heap etc. The second kind is raw data that a user stored in the file’s objects. The library uses a strategy internally to manage requests for these two kinds of file space and to track released free space. Free space of varied sizes is generated as the user manipulates the file’s objects.

The HDF5 library provides command line tools for users to examine the file’s contents and the distribution of file space, and to re-create a file with a specified file space handling strategy.

The first tool is *h5dump* by which the user can examine the file’s content. Look at the following output for an empty file *example.h5*:

HDF5 "example.h5" {

GROUP "/" {

}

}

Even though the file is empty, the library automatically creates the root group and allocates file space for metadata to describe itself. The file size at this point is the size of the library’s metadata.

If three datasets of different sizes are created in the file *example.h5*, *h5dump* will produce the following output:

HDF5 "example.h5" {

GROUP "/" {

DATASET "dset1" {

DATATYPE H5T\_STD\_I32LE

DATASPACE SIMPLE { ( 10 ) / ( 10 ) }

DATA {

(0): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

}

}

DATASET "dset2" {

DATATYPE H5T\_STD\_I32LE

DATASPACE SIMPLE { ( 30 ) / ( 30 ) }

DATA {

(0): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

(19): 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29

}

}

DATASET "dset3" {

DATATYPE H5T\_STD\_I32LE

DATASPACE SIMPLE { ( 50 ) / ( 50 ) }

DATA {

(0): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

(19): 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,

(35): 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49

}

}

}

}

The file size at this point consists of raw data and metadata.

The second command line tool is *h5stat*, which allows the user to find out the file space distribution for metadata, free space and raw data in more detail. See the following *h5stat* output and the corresponding file layout for *example.h5*:

Filename: ./example.h5

:

:

:

Storage information:

Superblock: 96

Superblock extension: 0

User block: 0

Object header (total/unused):

Groups: 40/0

Datasets: 816/432

Datatypes: 0/0

Groups:

B-tree/List: 872

Heap: 120

Attributes:

B-tree/List: 0

Heap: 0

Chunked datasets:

B-tree: 0

Shared Messages:

Header: 0

B-tree/List: 0

Heap: 0

Free space managers:

Header: 0

Amount of free space (in bytes): 0

:

:

:

Dataset storage information:

Total raw data size: 360

|  |  |  |
| --- | --- | --- |
| Metadata  (1944 bytes) | ??  (104 bytes) | Raw data  (360 bytes) |

File layout for *example.h5* (2408 bytes)

Note that the file at this point consists of 360 bytes of raw data for the three datasets as well as 1944 bytes of metadata such as superblock, object headers etc. The *h5stat* output seems to indicate that there is no free space in the file. However, the sum of all metadata and raw data does not equal to the file size of 2408 bytes. The discrepancy of 104 bytes is due to the amount of free space that is lost because of the library’s default file space handling strategy of not persisting free space at file closing.

Look at the following *h5stat* output and the corresponding file layout when one dataset (*dset2*) is removed from the file:

Filename: ./example.h5

:

:

:

Storage information:

Superblock: 96

Superblock extension: 0

User block: 0

Object header (total/unused):

Groups: 40/0

Datasets: 544/288

Datatypes: 0/0

Groups:

B-tree/List: 872

Heap: 120

Attributes:

B-tree/List: 0

Heap: 0

Chunked datasets:

B-tree: 0

Shared Messages:

Header: 0

B-tree/List: 0

Heap: 0

Free space managers:

Header: 0

Amount of free space (in bytes): 0:

:

Dataset storage information:

Total raw data size: 240

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metadata  (1400 bytes) | ??  (272 bytes) | Metadata  (272 bytes) | ??  (104 bytes) | Raw data  (40 bytes) | ??  (120 bytes) | Raw data  (200 bytes) |

File layout for *example.h5* (2408 bytes)

Note that the file at this point consists of two separate sections of raw data totaling 240 bytes and two separate sections of metadata totaling 1672 bytes. The three unknown sections of 496 bytes are free space sections that are not persisted at file closing due to the library’s default file space handling strategy. As can be seen from the above example, the removal of one dataset causes holes (fragments) in the file.

Finally, the third command line tool, *h5repack*, allows users to re-pack a file from an existing HDF5 file with a specified file space handling strategy. Look at the following *h5stat* output and file layout when *example.h5* is re-packed with a file space handling strategy of persisting free space at file closing:

Filename: out\_example.h5

:

:

:

Storage information:

Superblock: 48

Superblock extension: 88

User block: 0

Object header (total/unused):

Groups: 40/0

Datasets: 256/0

Datatypes: 0/0

Groups:

B-tree/List: 872

Heap: 120

Attributes:

B-tree/List: 0

Heap: 0

Chunked datasets:

B-tree: 0

Shared Messages:

Header: 0

B-tree/List: 0

Heap: 0

Free space managers:

Header: 117

Amount of free space (in bytes): 507

:

:

:

Dataset storage information:

Total raw data size: 240

|  |  |  |
| --- | --- | --- |
| Metadata  (1541 byte) | Free space  (507 bytes) | Raw data  (240 bytes) |

File layout for *out\_example.h5* (2288 bytes)

The above output indicates the existence of free space on top of metadata and the user’s raw data. Actually, the sum of raw data, metadata and free space equals to the size of the file.

The next section will give detailed explanation of the different file space handling strategies as well as the means to create an HDF5 file with the desired selection.