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 (such as superblock, object header, B-tree, heap, etc.), which the HDF5 library uses to describe itself and to identify objects (such as groups, attributes, datasets, etc.). The second kind is raw data that a user stores 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 a 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*, which the user can use to examine a file’s content. The following *h5dump* output examines the 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 (*dset1*, *dset2*, and *dset3*) of different sizes are created in the file *example.h5*, *h5dump* produces 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 metadata and raw data.

The second command line tool is *h5stat*, which allows the user to see, in detail, the distribution of file space among metadata, free space and raw data. The following *h5stat* output and the corresponding file layout show these details for the file *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  (1,944 bytes) | ??  (104 bytes) | Raw data  (360 bytes) |

File layout for *example.h5* (2,408 bytes)

Note that the file at this point consists of 1,944 bytes of metadata (such as superblock, object headers, and groups) and 360 bytes of raw data for the three datasets. The *h5stat* output seems to indicate that there is no free space in the file. However, the sum of metadata and raw data does not equal the file size of 2,408 bytes. There is a discrepancy of 104 bytes, which is due to the amount of free space that is lost because the library’s default file-space-handling strategy does not allow free space to persist when a file is closed.

The following *h5stat* output and the corresponding file layout show the details for the file *example.h5* when one dataset (*dset2*) is removed:

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  (1,400 bytes) | ??  (272 bytes) | Metadata  (272 bytes) | ??  (104 bytes) | Raw data  (40 bytes) | ??  (120 bytes) | Raw data  (200 bytes) |

File layout for *example.h5* (2,408 bytes)

Note that the file at this point consists of two separate sections of metadata totaling 1,672 bytes and two separate sections of raw data totaling 240 bytes. The three unknown sections, totaling 496 bytes, are free-space sections that do not persist 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 caused fragments in the file.

Finally, the third command line tool, *h5repack*, allows users to repack a file from an existing HDF5 file with a specified file-space-handling strategy. The following *h5stat* output and the corresponding file layout show what happens when the file *example.h5* is repacked with a file-space-handling strategy that allows the free space to persist at file closing (resulting in the file *out\_example.h5*):

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  (1,541 bytes) | Free space  (507 bytes) | Raw data  (240 bytes) |

File layout for *out\_example.h5* (2,288 bytes)

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

The next section provides a detailed explanation of the different file-space-handling strategies as well as the means to create an HDF5 file with the desired strategy.