RFC: h5stat tool

Elena Pourmal, Quincey Koziol, John Mainzer

March 5, 2006
Version 0.2

1. Purpose of this document

This is a proposal for a tool to report different statistics on an HDF5 file and on objects in the file.

2. Motivation

While working on the “Compact Group” feature, Quincey wrote a tool called `h5stat`.

The tool iterates through the objects in the file and returns information on the file hierarchy and on the objects that constitute the hierarchy. It also returns some statistics on structural metadata in the file such as sizes of groups’ and datasets’ object headers, and free space in the headers. Appendix A contains an example of the statistics reported by `h5stat`.

The current implementation of `h5stat` has some limitations. In particular, the tool *does not gather statistics about the space taken by the internal file structures like local and global heaps and B-trees.* It also does not return some useful information on datatypes (for example, how much space named datatypes take in the file, or if a compound datatype is packed, what is a maximum number of members in a compound datatype, etc.) and on datasets (such as space taken by B-tree that is used for chunking storage, compression ratios, external storage, chunk sizes, applied filters, unlimited dimension, etc).

The tool also does not provide options to display particular information requested by a user (for example, information about groups or datasets only, or information about raw data vs. structural metadata only).

We propose to enhance the current tool to address the current limitations.

3. Use cases

3.1. A user has several ways to organize data in an HDF5 file. He/she would like to choose the structure of the file with a minimum size of structural metadata.

3.2. A user would like to know how many datasets in a file use external storage.

3.3. An application developer would like to perform performance analysis of his tool. He needs to select from an archive a set of “similar” HDF5 files to gather performance statistics.

3.4. A developer would like to move his application to use HDF5 file format instead of a binary format used for a long time in his organization. He has to demonstrate to the management that file in the HDF5 file format has an acceptable storage overhead.

3.5. An HDF5 library developer is working on a new library feature/file format change. Developer would like to know how much storage overhead this feature introduces in the file.

3.6. An HDF5 library developer would like to analyze HDF5 files gathered from an application’s use of the HDF5 library and determine if the structural metadata overhead can be reduced.

6. Proposed statistics

In this section we will describe statistics that will be reported by h5stat. Statistics marked with * are available in the current implementation. Since the proposed reported information may be overwhelming, it will be useful to have a several levels of verbosity for each HDF5 object described below.

6.1. File Object

6.1.1. File structural metadata (general)

- Total file size
- Actual HDF5 size (= ‘End of File Address’ – ‘Base Address’)
- File “*version*” number (*Note: we do not have file version number but only versions of the objects; for standardization and archiving purposes we have to come up with the versioning mechanism that will allow to identify the earliest version of the library that can read the entire file*; *the same comment may apply to an individual objects in the file*)
- Size of structural metadata (total/free). This includes sizes of all
  - object headers
  - global heap (size of the header only)
  - local heaps (for groups)
  - B-trees (for groups and chunked datasets)
- Fragmentation of
  - Objects headers (number of fragments and its distribution)
  - Local heaps (number of contiguous heaps and number of discontinuous heaps)
  - Global heap (if global heap is contiguous or discontinuous)

6.1.2. File structural metadata (specific)

- Size of the user block
- Information from Super Block
- Size of super block
- Group Leaf Node and Internal Node K numbers
- Size of offsets & lengths in the file
- Version #'s of things
- File Driver information if stored

6.1.3. File hierarchy

- Number of unique groups including Root Group*
- Number of unique datasets*
- Number of unique datatypes*
- Number of unique links *
- Number of soft links*
- Maximum* and minimum number of links to an object
- Maximum* depth of hierarchy
- Number of cycles in the graph
- Maximum and minimum cycles lengths
- Number of objects "outside" graph structure (for example named datatype that is used by some datasets but there is no path to it),

6.2. Group Object

6.2.1. For all groups

Structural metadata

- Total size of structural metadata for all unique groups in the file (total/free), which includes sizes of groups' headers, B-trees and symbol tables.
- Groups object header size (total/free)
- Fragmentation of group object headers (number of fragments and its distribution)
- Sizes of B-trees and % of occupancy
- Sizes of symbol tables (total/free)
- Symbol tables fragmentation (number of contiguous and discontinuous symbol tables)
- % of space in the group objects' headers occupied by attributes

Other info

- Maximum* and minimum number of objects in a group
- Number of small groups (specified by a user, may be a list of small groups)
- Group bins (default will be current log distribution*; number of bins and associated values specified by a user)
- Groups' bins for number of attributes

- Groups' bins for sizes of attributes data

6.2.2. For a particular group

Structural metadata

- Group object header size (total/free)
- Fragmentation of group object header
- Sizes of B-trees and % of occupancy
- Sizes of symbol table (total/free)
- % of space in the object header occupied by attributes

Other info

- Total size of attribute data
- Number of objects in the group
- Number of attributes

6.3. Dataset Object

6.3.1. For all datasets

Structural metadata

- Total size of structural metadata for all datasets in the file (total/free), which includes sizes of datasets' object headers and sizes of B-trees for chunked datasets.
- Fragmentation of dataset object headers
- Sizes of raw data (total/free)*
- Total size of structural metadata for all datasets in a group or list of groups (total/free) which includes sizes of datasets' object headers and sizes of B-trees for chunked datasets

Other info

- Datasets' bins by number of elements (regardless of rank)
- Datasets' bins by raw data storage size
- Datasets' bins by dataspace rank and dimensions
- Datasets' counts by layout (compact*, contiguous*, chunked*, external)
- Datasets' bins by compression ratio
- Datasets' bins by number of attributes
- Datasets' bins by sizes of attributes
- Datasets' bins by chunk dimensions
- Datasets' bins by datatype used

6.3.2. For a particular dataset:

Structural metadata

- Size of structural metadata for a dataset (total/free) , which includes sizes of dataset's object header and size of B-tree if dataset is chunked.
- Fragmentation of object header (number of fragments and their sizes)
- Size of raw data (total/free)*
- % of space in the object header occupied by attributes

- % of space in the object header occupied by datatype
- % of space in the object header occupied by dataspace (for completeness of the picture ☺)
- number of header messages and their sizes

Other info

- For compressed/chunked dataset:
  - Compression ratio if compressed
  - Compression or filter applied
  - Number of written chunked
  - Chunk size
- Number of attributes
- Total size of attribute data

6.4. Datatype Object

6.4.1. For all datatypes

Structural metadata

- Total size of space in the dataset object headers occupied by the datatype description
- % of dataset object headers size occupied by datatype description

Other info

- Number of unique datatypes used by datasets*
- Number of datasets that use a particular datatype
- Number of named datatypes

6.4.2. For all named datatypes

Structural metadata

- Size of object headers for all named datatypes in the file (total/free)
- Fragmentation of named datatype header objects

6.4.3. For a particular named datatype

Structural metadata

- Size (or sum of) of object header(s) for named datatype (total/free)
- Fragmentation of object header

Other info

- Number of datasets that use this named datatype

6.5. Link Object

- Number of unique links
- Number of hard links
- Number of soft links
- Number of external links

7. New functions needed to get the required information

The h5stat tool will require additional private HDF5 API routines to be added to the HDF5 library, in order to gather the information described above. The new APIs routines may become public in the future if we find that those routines will be broadly useful to applications other than the h5srtat tool, have low maintenance cost and do not depend on the internal library structures.

Here is an outline of proposed APIs to collect information about:

1) Object headers

We need two API calls returning the following information:

a) total object header size
b) # of internal (i.e not attribute) messages
c) # of attribute messages
d) # of free space messages
e) # of fragments in object header (i.e. # of continuation blocks)
f) total internal message size
g) total attribute message size
h) total free space
i) list of internal message sizes
j) list of attribute message sizes
k) list of free space message sizes

The first API will return items a - h. The second will load items i - k into buffers provided by the caller. Input for both APIs will be an object ID.

2) Local heap

We will need two API calls returning the following information:

a) Heap size
b) Whether the heap is fragmented (i.e. is the data section adjacent to the header).
c) Total entry size
d) Total free space
e) # of entries
f) # of free space entries
g) list of entry sizes
h) list of free space entry sizes

The first API will return items a - f. The second will load items g - h into buffers provided by the caller. Input to both APIs will be the ID of the group containing the local heap.

3) Global heap:

API should be similar in concept to the local heap. We must review global heap code before deciding on information returned.

4) B Trees

We will need an API call returning the following information:

a) Total space used by the B Tree
b) Node size
c) Entries per node
d) Depth
e) % of entries used

Input to an API will be ID of the group or chunked dataset.

8. Syntax

h5stat <flags> *filename.h5*

Flags and corresponding outpare described in the table below.

Flag(s)	***Description***
-h, --help	Displays information about the h5stat tool
-all (default)	Displays all statistics
-F, --filemetadata	Displays all statistics on the file as described in the subsections 6.1.1. and 6.1.2
-f , --filehierarchy	Displays file's hierarchy statistics as described in the subsection 6.1.3
-G, --groupmetadata <group_name>	Displays structural metadata statistics on the groups or a group as described in the sections 6.2.1 and 6.2.2
-g, --grouphierarchy <group_name>	Displays all statistics on the groups or on a group as described in the sections 6.2.1 and 6.2.2 "Other info"
- D , - -dsetmetadata <dataset_name>	Displays all structural metatdata statistics on the datasets or a datasets as described in the section 6.3.2
-d , --dset <dataset_name>	Displays all statistics on the datasets or a datasets as described in the section 6.3.2 "Other info"
- T , - -dtypemetadata <datatype_name>	Displays all structural metatdata statistics on the datatypes or a datatype as described in the sections 6.4.1, 6.4.2, and 6.4.3 or 6.4.3

-t <datatype_name>	Displays all statistics on the datatypes or a datatupe as described in the sections 6.4.1, 6.4.2 and 6.4.3 or 6.4.3 “Other info”
-link	Displays all statistics on the links as described in the section 6.5

9. Distribution

The source code for the tool will be in the `tools/misc` directory in the main HDF5 source distribution. It will be built along with other tools in the same directory and installed under `bin` subdirectory of the installed distribution.

Appendix A

Example of current h5stat output

```
Filename: test.h5
File Hierarchy:
        # of unique groups: 50630
        # of unique datasets: 29895
        # of unique named dataypes: 0
        # of unique links: 0
        # of unique other: 0
        Max. # of links to object: 1
        Max. depth of hierarchy: 5
        Max. # of objects in group: 10510
Object header size: (total/free)
        Groups: 2430240/405040
        Datasets: 54903560/2391600
Small groups:
        # of groups of size 1: 46043
        # of groups of size 2: 4583
        # of groups of size 3: 1
        Total # of small groups: 50627
Group bins:
        # of groups of size 1 - 9: 50627
        # of groups of size 1000 - 9999: 2
        # of groups of size 10000 - 99999: 1
        Total # of groups: 50630
Dataset dimension info:
        Max. rank of datasets: 1
        Dataset ranks:
                # of dataset with rank 1: 29895
```

```
1-D Dataset info:
        Max. dimension size of 1-D datasets: 63381
        Small 1-D datasets:
                # of dataset dimensions of size 1: 2032
                # of dataset dimensions of size 2: 883
                # of dataset dimensions of size 3: 609
                # of dataset dimensions of size 4: 470
                # of dataset dimensions of size 5: 367
                # of dataset dimensions of size 6: 271
                # of dataset dimensions of size 7: 233
                # of dataset dimensions of size 8: 151
                # of dataset dimensions of size 9: 139
                Total small datasets: 5155
        1-D Dataset dimension bins:
                # of datasets of size 1 - 9: 5155
                # of datasets of size 10 - 99: 5405
                # of datasets of size 100 - 999: 9713
                # of datasets of size 1000 - 9999: 8384
                # of datasets of size 10000 - 99999: 1238
                Total # of datasets: 29895
Dataset storage info:
        Total raw data size: 737671303
Dataset layout info:
        Dataset layout counts[COMPACT]: 0
        Dataset layout counts[CONTIG]: 0
        Dataset layout counts[CHUNKED]: 29895
Dataset datatype info:
        # of unique datatypes used by datasets: 2
        Dataset datatype #0:
                count = 25312, desc. size = 1626, data elmt size = 124
        Dataset datatype #1:
                count = 4583, desc. size = 1182, data elmt size = 84
        Total dataset datatype count: 29895
```