
USING LATEST VERSION OF FILE FORMAT IN H5REPACK

Peter Cao
May 31, 2007

1. INTRODUCTION

Combining the h5repack tool with the “use latest format” feature introduced in HDF5 Release 1.8.0, we can take advantage of several HDF5 1.8 file format tuning features when we repack an HDF5 file. By default, h5repack repacks a file based on the default settings of the library. Since h5repack is always working with a complete source file, we can discover what is in the file that is to be repacked, and then use selected new format features to repack the file more efficiently.

2. COMPACT GROUP SIZE

In HDF5 Release 1.8.x, groups can be stored in file in compact or indexed formats. (See the introductory paragraphs of the H5G API page in the Release 1.8.0 *HDF5 Reference Manual* for a discussion of compact and indexed groups: http://www.hdfgroup.uiuc.edu/HDF5/doc_dev_snapshot/H5_dev/RM/RM_H5G.html)

In the compact format, links are stored as messages in the group’s header. The tuning parameter *max_compact* sets the maximum number of links to store as header messages in the group header before converting the group to the indexed format. Groups that are in the compact format and exceed this number of links are automatically converted to the indexed format.

In the indexed format, links are stored in an enhanced B-tree in the group’s local heap. The minimum indexed tuning parameter, *min_dense*, sets the minimum number of links to store in the index before converting the group to the compact format. Groups that are in the indexed format and fall below this number of links are automatically converted to the compact format.

For example, the number of anticipated group members is one factor in determining the space allocated for an object header. If 90% of the groups in a file have 2 members, setting the maximum compact size to 2 may prevent the allocation of a lot of unnecessary space in the group object headers. In another case, if most of the groups have 9 members, these groups would be converted to indexed groups with the default settings. That might degrade performance, but setting the maximum compact size to 9 will ensure that these groups remain compact.

By default, the maximum compact size is 8 and minimum index size is 6, but both can be reset by applications. These two values can be the same, but they should normally be different to minimize the risk of having a group that is right at the threshold frequently switch back and forth between compact and indexed the formats. Consider a group with 8 members that repeatedly

have a new member added then deleted. If both values were set to 8, the group would change to indexed format every time the new group was added and switch back to compact format each time it is deleted. This would create unnecessary thrashing that could impair applications performance. If the minimum indexed value were set to 6, the group would be changed to indexed format at the outset and none of the unnecessary switching each time the new group was deleted would take place.

3. SHARED OBJECT HEADER MESSAGE SIZE

Shared object header messages present a similar opportunity to take advantage of an HDF5 file optimization.

The shared object header message feature provides a means of saving significant space in an HDF5 when there is repeated use of identical large header messages in a file. For example, if a file contains 10,000 datasets with the same complex dataspace definition or the same large attribute definition, this feature makes it possible to store one copy of each header message that is used repeatedly and store just a pointer each time it is used.

The shared object header message settings can make use of two sets flags: the minimum message size and header message types. The source file can be analyzed for repeated header messages that occupy a lot of space in the file and could be shared. Then sizes of those shared message can be analyzed and we can set the types of messages to share and the minimum size of shared messages, maximizing space saved in the file.

4. NEW h5repack FLAGS

4.1 FLAG FOR LATEST FORMAT

The first flag invokes the use of the latest HDF5 format features in the repacked file:

-L, --latest Use latest version of file format to create groups, datasets and datatypes

4.2 FLAGS FOR HEADER MESSAGES

The following flags can be used to tune the feature settings to take maximum advantage of the format options. Without these flags, a file will be repacked with the HDF5 library default settings:

-compact=<size> Set the maximum number of links to store as header messages in a group (compact format)

-indexed=<size> Set the minimum number of links to store in the indexed format

4.2 FLAGS FOR SHARED MESSAGES

-ssize=<size>[:<msg>] Set the shared object header message minimum size
<size> is the minimum size of object header message that will be stored as a shared message.

<msg> specifies the type(s) of header messages to be share and will be one or more of the following: dspace (dataspaces), dtype (datatypes), fill (fill values), pline (Filter pipeline) and attr (attributes). If <msg> is not specified, <size> applies to all messages.

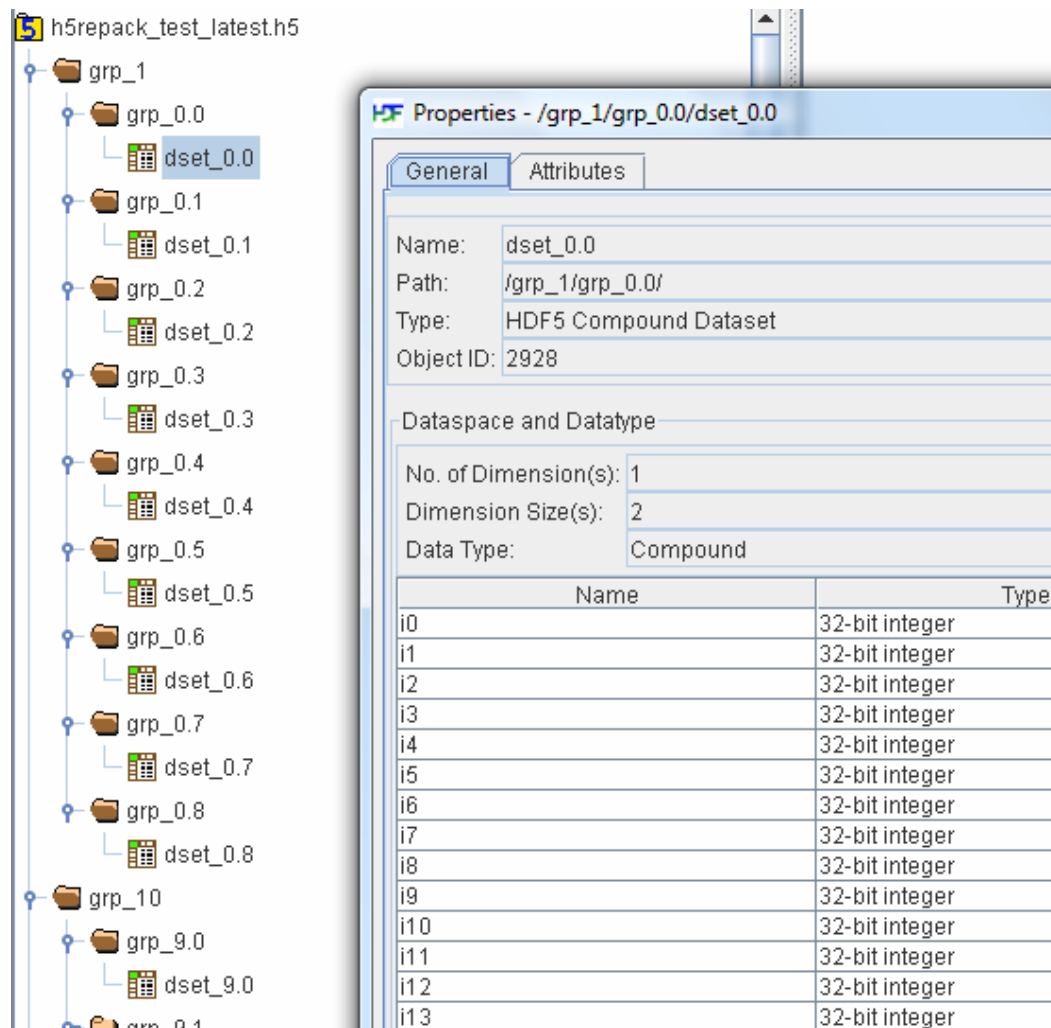
Examples:

-ssize=<100> Sets the minimum size for all shared messages to 100 bytes.

-ssize=10:fill Sets the minimum size of shared fill values to 10 bytes.

4. A SIMPLE TEST

The testing file was created with HDF5 1.6.5. The root group of the file contains 100 groups. Each group at the root contains 9 sub groups. Each sub group contains one compound dataset. The compound dataset has 40 members of 32 bit integer. The file looks like



Below are some test results with different options. The new file is over 10 times smaller with "--latest -compact=9 -ssize=50:dtype" option.

h5repack options	File size in bytes
Original file	3,167,264
--latest	2,267,346
--latest -compact=9	2,162,254
--latest -compact=9 -ssize=50:dtype	299,882