RFC: Merging Named Datatypes in H5Ocopy

Neil Fortner

Quincey Koziol

This RFC proposes adding a new feature to H5Ocopy to allow matching named datatypes used by datasets and attributes in the source file with already existing named datatypes in the destination file. When such a match is found, the dataset or attribute will use the existing named datatype in the destination file. This will make it easier to maintain the shared nature of named datatypes when copying objects between files.

Introduction

Named datatypes can be a powerful feature in HDF5. They can be used to share a single datatype description among multiple datasets, saving space, and to assign a name to that datatype within the HDF5 group structure. However, problems can occur when using *H5Ocopy* on named datatypes or datasets using named datatypes.

When copying a dataset that uses a named datatype between files, the library does not look for a matching named datatype in the destination file, it simply creates a new named datatype in the destination file without any links to it (an anonymous named datatype). This means that, when copying multiple datasets in separate calls to *H5Ocopy*, a new named datatype is created for each *H5Ocopy* call. While it is possible to have the copied datasets share the same datatype by copying all the datasets in a single call to *H5Ocopy* by copying a parent group, this is not always possible.

For example, imagine that a user has an application that automatically creates many data files, each with many datasets that all use a single named datatype. At the end of a project, the user wants to merge all of these files into a single file. There is currently no way to have all of the datasets in the combined file use the same named datatype, short of manually recreating each dataset in the combined file.

This RFC proposes adding a new feature to *H5Ocopy*, by means of a new flag for use with *H5Pset_copy_object*, which directs the library to, when copying a dataset that uses a named datatype to a different file, search for matching a named datatype in the destination file and, if found, have the copied dataset use the found datatype.

Approach

We will expose this functionality by means of a new bitflag, *H5O_COPY_MERGE_NAMED_DTYPE_FLAG*, which can be added to the *copy_options* parameter for *H5Pset_copy_object*. When this option is set, *H5Ocopy* will, when it first encounters a dataset or

attribute using a named datatype, search the destination file for named datatypes and build a temporary list in memory of all named datatypes found. It will then check if that list contains a datatype equal to the datatype of the source object, and if so, modify the copied object so that it uses the found named datatype as its datatype. When later datatsets and attributes using named datatypes are encountered, the library will again check if the list contains a matching datatype, and will update the list if a new named datatype is created in the destination file as a result of the copy.

When the library encounters a named datatype in the source file, it will similarly search for a matching named datatype in the destination file. If a match is found, the library will simply create a hard link in the destination file to the found datatype. If a match is not found, the library will copy the named datatype normally and add it to the temporary list of named datatypes in the destination file.

Example

This example shows how to enable this feature for use with *H5Ocopy*.

```
hid_t ocp_plist_id;

ocp_plist_id = H5Pcreate(H5P_OBJECT_COPY);
status = H5Pset_copy_object(ocp_plist_id, H5O_COPY_MERGE_NAMED_DT_FLAG);
status = H5Ocopy(file1_id, src_name, file2_id, dst_name, ocp_plist_id,
H5P_DEFAULT);
```

Revision History

August 25, 2011: Version 1 circulated for comment within The HDF Group.

August 29, 2011: Version 2 circulated for comment within The HDF Group.