RFC: Caching Files Opened Through External Links

Quincey Koziol

External links in HDF5 provide transparent access to objects in other HDF5 files. However, when an object opened through an external link is closed, the file that the object is in is closed also, causing a performance problem if that object is accessed repeatedly. This document describes a solution to this problem.

1 Introduction

External links in HDF5 files provide a transparent mechanism for accessing objects in other HDF5 files, without an application being aware that the object is not in the current file. However, when an object that was opened through an external link is closed, it closes the external file and releases all cached information about that file. This can cause problems if the same external link, or another external link into the same file is accessed again shortly afterward.

This RFC describes a mechanism for keeping files opened through external links open for further access.

2 Approach

Files opened through an external link ("child files") will be cached in a data structure in the file that contains the external link (the "parent file"). By default no child files will be cached in the parent file, this feature must be enabled with a new file access property list API function (described below). Child files can be closed and removed from the parent file with another new API routine (also described below).

3 New API Routines

In this section, the new API routines are described.

3.1 File Access Property List Routines

As usual, the file access property described below is inherited by external files, allowing multiple levels of child file caching to occur.

3.1.1 H5Pset_elink_file_cache_size

This routine takes a file access property list ID and an unsigned value to set the number of opened child files to cache. The default value is zero.

3.1.2 H5Pget_elink_file_cache_size

This routine takes a file access property list ID and a pointer to an unsigned value, which is set with the number of opened files to cache.

3.2 File Routines

3.2.1 H5Fclear_elink_file_cache

This routine takes a file ID, and closes all child files that are currently cached for the file ID. Closing a file will also close any open child files (and thus calling H5Fclear_elink_file_cache will recursively clean up a multi-level hierarchy of cached open child files).

4 Implementation

We should probably have a small LRU list of open files for each “shared” file struct in the library. Lookups in the file aren’t necessary, since the underlying file will be matched when it is re-opened again through the external link.

The implementation will need to protect against cycles in the external links (possibly in a similar manner to how mounted files are protected against cycles).

5 Example code

A short example of setting up external file caching and its use:

```
#include “hdf5.h”

main()
{
    hid_t fapl;
    hid_t fid;
    hid_t dsid;

    /* Set up external link file caching, retaining 16 open files */
    fapl = H5Pcreate(H5P_FILE_ACCESS);
    H5Pset_elink_file_cache_size(fapl, 16);

    /* Open a file, turning on external file caching for it */
    fid = H5Fopen(“filename.h5”, H5F_ACC_RDONLY, fapl);

    /* Close the FAPL, we’re done with it */
    H5Pclose(fapl);

    /* Open a dataset through an external link */
    dsid = H5Dopen2(fid, “/path/to/dataset”, H5P_DEFAULT);

    /* Do stuff with the dataset */

    /* Close the dataset */
    H5Dclose(dsid);
    /* Normally, the child file would be closed, but with caching turned on,
       it’s not, making the next dataset open fast */
    dsid = H5Dopen2(fid, “/path/to/dataset”, H5P_DEFAULT);

    /* Do something more with dataset */

    /* Close dataset */
    H5Dclose(dsid);

    /* Release all the cached open child files for the parent file */
    H5Fclear_elink_file_cache(fid);

    /* Reopen dataset, which is costly again, since the cached file is closed */
    dsid = H5Dopen2(fid, “/path/to/dataset”, H5P_DEFAULT);

    /* Do something more with dataset */

    /* Close dataset */
    H5Dclose(dsid);

    /* Close file */
    H5Fclose(fid);
}
```

6 Further Ideas

We might want some way to query the number of cached files. Possibly some way to track the open file cache’s effectiveness also. But, that functionality probably can wait for later when we know better what application developers using the basic routines above would like to query.

7 Recommendation

We should implement this simple addition to the library’s API and functionality, to speed up frequent access to objects in files reached through external links.

Revision History

November 4, 2010: Version 1 circulated for comment within The HDF Group.

November 8, 2010: Version 2 revised after internal comments. Circulated to hdf-forum and other external HDF5 users.