Support external link caching feature for HDF5 tools

Jonathan Kim

This document is about using the new HDF5 APIs which were added to improve performance for frequent accessing same objects through external links. These APIs will be used for HDF5 command line tools and a new option will be introduced to give user optional control.

# Introduction

New HDF5 APIs were added to HDF5 library for the purpose of improving performance when accessing same objects repeatedly through external links.

HDF5 tools’ performance can be improved by utilizing the APIs for users who need to access objects repeatedly through external links with HDF5 tools. So this feature will be implemented to HDF5 command tools one by one as a default feature since this only provide toward performance benefit and no backward compatibility issues.

This RFC also introduce a new optional argument for controlling file caching number among HDF5 command-line tools.

For a reference, the API “H5Pset\_elink\_file\_cache\_size (fapl, Num)” will be used. This routine takes a file access property list ID and an unsigned value to set the number of opened child files to cache.

Refer to the “RFC: Caching Files Opened through External Links” for details about the HDF5 APIs.

# Target command line tools

Currently there are 4 tools that are support accessing object through external link.

Those are ***h5diff***, ***h5dump***, ***h5ls*** and ***h5copy***.

In near future ***h5repack*** will support external link as well.

# Use cases

Assumption:

There are HDF5 files with several objects and there are other HDF5 files with lots of external links to the objects in the other HDF5 files. This means that accessing the objects through external links will cause frequent and repeated access to the same objects when user runs HDF5 tools on the external link side HDF5 files.

Following symbolic links option is turned on with the HDF5 tools listed.

1. User wants to use ***h5diff*** tool on the above HDF5 files containing lots of external links for comparing the files or groups.
2. User wants to use ***h5dump*** tool on the above HDF5 files containing lots of external links for displaying HDF5 files’ contents.
3. User wants to use ***h5ls*** tool on the above HDF5 files containing lots of external links for listing objects or viewing object data with h5ls.
4. User wants to use ***h5copy*** tool on the above HDF5 files containing lots of external links for copying HDF5 files or groups.

# A new argument for HDF5 tools

## --set-elink-cache=N

The ‘N’ is a number equal or bigger than 0, which sets the maximum number of retainable open files to cache.

If 0 is used, it will be same as turn off the caching feature (same as current situation).

This argument will affect performance only when used with ‘—follow-symbolic’ option over external links.

HDF5 tools will use the API to utilize external link file caching feature as default action and default N would be 16, so user doesn’t need to bother setting anything number to get the performance benefit.

This option is provided to give certain power-users to tinker with the N for exceptional cases.

# Test

Since this is performance updating, there is not visual change. So test will be performed to compare time measurements before and after implementing this feature.

The test status and results will be recorded into JIRA’s task comment, since we don’t build tools with previous source.

The test files will include small and big numbers of external links in ways our customer uses heavily.

If this test shows the performance is better, it will be considered as success.

# Acknowledgements

This work was supported by a customer of The HDF Group.

# Revision History

|  |  |
| --- | --- |
| *April 1, 2011:* | Version 1 circulated for comment within The HDF Group. |
| June 3, 2011: | Version 2 circulated for comment |
|  |  |
|  |  |
|  |  |
|  |  |