HDF5 VOL tests refactoring plan

Jordan Henderson

This document details a plan for integrating the external HDF5 VOL tests back into the HDF5 library and refactoring them for maintainability.

Revision History

Version Number	Date	Comments
v1	Apr. 03, 2023	Version 1 drafted.

Contents

1 Introduction and overview

During the development of HDF5's Virtual Object Layer abstraction layer, and related connectors, such as the HDF5 DAOS VOL connector, it became apparent that HDF5's existing tests weren't suitable for testing with VOL connectors at the time. Due to issues such as testing of HDF5 file format-specific notions and other functionality that a VOL connector may not support, it was determined that either these tests would need to be updated or new tests would have to be written.

While consideration was given to updating HDF5's existing tests to be compatible with VOL connectors, two problems were noted. First and foremost, the necessary changes of separating and compartmentalizing file format-specific tests and testing of possibly unsupported functionality would have been pervasive. Second, an ordering issue existed in that HDF5 had to be built before a VOL connector could be built. Since HDF5's tests would have been run during the building of the library, and since binaries for HDF5's test programs don't get installed with the library, a user would have been faced with the awkward problem of needing to re-use an HDF5 build directory for testing purposes when building a VOL connector. If the VOL connector were being built with a version of HDF5 already installed on the system, a build directory wouldn't even be available.

Due to time constraints when developing the HDF5 DAOS VOL connector, it was decided that a new, relatively small test suite would be written in order to cover basic HDF5 functionality using just the public API. Then, a subset of tests from HDF5 which cover more advanced functionality of the library would be copied verbatim and quickly hacked up to remove all testing of problematic functionality. These tests currently exist in the HDF5 VOL tests repository. While these tests initially served their purpose well, concerns arise with their maintainability since they exist outside of the main HDF5 library and contain a moderate amount of duplicate test code from HDF5. Indeed, there are occasional issues with using these tests as they gradually became out of sync with changes in HDF5 over time. To address concerns about these tests' maintainability and future usability, an initial plan for integrating them back into the HDF5 library has been developed and is detailed in this document.

2 Test integration effort

The sections 2.1-2.5 outline the general plan for integrating the tests contained in the HDF5 VOL tests repository back into the library. A new feature branch has been created in the HDF5 repository for this purpose and is under active development.

2.1 Structure of tests in the HDF5 library

The existing VOL tests repository consists of two different types of tests, the newly-written HDF5 public API tests and a set of heavily modified tests copied directly from the HDF5 library's existing tests. The plan for this test integration effort is to create new `API` directories under HDF5's existing `test` and `testpar` directories which will contain the newly-written serial and parallel tests, respectively. The tests that were copied from HDF5 will not be moved into these directories, as that would duplicate existing tests and add to the burden of maintaining these tests. Instead, these tests will simply be built from the library's existing test files as usual. However, note again that several modifications were made to these tests to make them function correctly when used with a VOL connector. These modifications must be accounted for by updating the existing tests in the library to function in the same manner correctly.

Once the structure for these tests has been set up and the tests have been moved back into the library, the library's CMake and Autotools build code will then be updated to build all of the test files under `test/API` into a single executable, and all of the test files under `testpar/API` into a single executable. Due to issues that occur when mixing static and shared libraries in the context of HDF5 and VOL connectors, the build code will, where possible, prefer only to build versions of these test executables which link against a shared HDF5 library. If only the static HDF5 library is built, these test executables will still be built since they are useful independently of any HDF5 VOL connectors. In this case, though, some provision should be made, either at build time or during the execution of the test programs, for warning users that the executables should not be used for testing VOL connectors.

2.2 Test driver program

The existing HDF5 VOL tests repository contains code for a test driver program that uses Kitware's System Library. This test driver program is primarily useful when testing VOL connectors that require some form of server executable in order to function. Since the HDF5 DAOS VOL connector uses this test driver program, the intention is to temporarily move this code into HDF5 under the new `test/API` directory. However, future discussion about the driver program's eventual location will be necessary since including it in HDF5 for a single optional purpose is not ideal. Integrating the driver as a git sub-module is one approach, but sub-modules have historically been a pain to work with, so perhaps using CMake functionality to bring the driver code in at build time would be better. Though, this still leaves the question of how to support the same functionality for Autotools builds.

2.3 Testing of VOL connectors

The existing process for testing an HDF5 VOL connector that is external to the library looks similar to the following:

1. Build HDF5 and install it somewhere on the system
2. Build the VOL connector using the HDF5 installation
3. Build the HDF5 VOL tests
4. Set the `HDF5_VOL_CONNECTOR` and `HDF5_PLUGIN_PATH` environment variables and run the HDF5 VOL tests

To streamline this process a bit, the plan is to add a new option to HDF5's build code which will allow the user to specify a VOL connector or set of connectors to be built and installed alongside the HDF5 library. In this manner, the process of building VOL connectors can be placed between the building of the library and the building/running of the library's tests.

For the CMake code, it should be enough to allow users to specify one or more URLs that point to VOL connector code (such as git repository URLs). After the HDF5 library has been built, the CMake code would then attempt to fetch code from the specified URLs using CMake's FetchContent functionality and build each available VOL connector. It would be nice to support the same functionality for HDF5's Autotools build code, but the process will likely not be as straightforward, so CMake will be used as a proof-of-concept before revisiting the support in the Autotools code.

2.4 Test installation option

As a convenience option for users, the library's CMake and Autotools build code will be updated with a new option that will allow installation of the test executable binaries onto the system where other HDF5 executables (such as `h5dump`) are installed. This option will enable users to maintain a workflow similar to their current one by combining the building of HDF5 and the VOL tests together and making the test executables available after HDF5 has been built and installed. It will also allow those responsible for building HDF5 on a system to ensure the test binaries are available for future use in testing VOL connectors.

2.5 Configuring a default VOL connector

While discussing approaches for building VOL connectors as part of the HDF5 library, it was realized that it could be helpful to have an option for configuring the default VOL connector for HDF5 to be something other than the native VOL connector. This would function in an identical manner to setting the `HDF5_VOL_CONNECTOR` environment variable, where the default VOL connector for a File Access Property List would be replaced with the VOL connector specified as the new default connector. Depending on how the VOL connector obtains any parameters for its configuration, an HDF5 application using an HDF5 library configured in this way wouldn't have to be modified in any way and no environment variables would need to be set in order to use the particular VOL connector chosen. No concrete plans for such an option have been developed yet, but this remains an interesting option that could be supported in the future.

3 Test refactoring

Sections 3.1-3.4 detail work that will need to be performed once the VOL tests have been integrated into the library in order to bring those tests back into alignment with HDF5's other tests. Since this work is somewhat open-ended, these sections are generally ordered by the estimated completion time.

3.1 Minor Rename

The current tests in the VOL tests repository make several references to VOL connectors, with the test executable names even being prefixed with "h5vl". Since these tests are written to test the public HDF5 API and are helpful independently of any VOL connectors, it is suggested that these tests should be renamed to have a prefix of "h5_api" instead of "h5vl" and that most references to "VOL" should be replaced with references to "API" where appropriate.

3.2 Integration with HDF5 testing frameworks

To avoid a substantial future re-write of testing code, these tests were originally written to make use of HDF5's "h5test" testing framework and macros. Since that test functionality was private to the HDF5 library, the relevant macros, variables, etc. were copied into the VOL tests repository for external use there (https://github.com/HDFGroup/vol-tests/blob/hdf5-1.14.0/vol_test.h). Some of the test functionality was also updated within the VOL tests repository to do valuable things, such as tracking test statistics that the VOL tests print out after running. Meanwhile, the same test functionality was also updated within HDF5, such as by converting the testing macros into a `do .. while()` form that works better with code formatters. Integrating the VOL tests back into HDF5 means that all changes must be synchronized between the two versions of the testing framework functionality. This will include: ensuring that all the testing macros used in the VOL tests end in semi-colons, updating HDF5's testing macros to include the test statistic tracking for use by the VOL tests, removal of any duplicated variables, test utility functions, etc., from the VOL tests and so on.

Consideration has also been given to integrating the VOL tests with HDF5's "testhdf5" framework since it provides excellent testing features, such as running a specified subset of tests. However, that testing framework is considered chiefly legacy at this point, so for now, the tests will only be integrated with the "h5test" framework. There have also been discussions around unifying the "h5test" and "testhdf5" frameworks, which would include these tests, but nothing has come of those discussions yet.

3.3 VOL connector capability flags

When these tests were written, HDF5 had no support for determining what kind of functionality a VOL connector supported. Due to this, a VOL connector with only partial support for the HDF5 API would experience many failures when tested against the VOL tests. Therefore, in the HDF5 1.14.0 release, a new API routine, `H5Pget_vol_cap_flags`, was added to enable querying a VOL connector about its supported HDF5 functionality. The routine returns a set of "capability flags" that a VOL connector must define in order to inform the caller about what functionality is supported and can safely be tested.

The VOL tests repository was updated to perform basic checking of a VOL connector's defined capability flags before running certain portions of the newly-written API tests. Still, this work hasn't been fully completed yet. Further, no checking of capability flags currently exists within the library's existing tests, so those tests will need to be updated as well. The VOL tests repository also contains a header file, vol_tests_disabled.h, which defines several macros to disable tests that proved to be problematic for particular VOL connectors. With support for VOL connector capability flags in place, most of the macros in this header file should be able to be removed in favor of surrounding those tests with checks for supported functionality before running them. It is yet to be determined if additional VOL connector capability flags need to be added to HDF5 to support eliminating this header file. See the VOL feature flag RFC for more details about the feature flags.

3.4 Splitting of test functionality

In order to make the VOL test repository's copies of existing HDF5 tests run and pass when used with a VOL connector, the majority of them had to be updated to disable large portions of the code that tested HDF5 file format-specific things, such as expected sizes of files after performing a particular set of operations. Since we still need to disable these types of tests when running with VOL connectors other than HDF5's native VOL connector, the majority of HDF5's existing test code will need to be updated to handle this situation by checking the VOL connector being used, rather than by compiling out native VOL connector-specific tests. Initially, this test integration plan will focus specifically on updating just the HDF5 tests that were copied into the VOL tests repository. Once this has been accomplished and the tests have been verified to work correctly with external VOL connectors, additional HDF5 tests can be updated and made available for testing with VOL connectors as necessary. This will likely be a large and open-ended task, so it will be investigated as time permits.