h5sh Documentation

Release 0.1.1

Seth Robert Johnson

Dec 13, 2021

Contents

Abstract

This project provides a shell-like interface for navigating and extracting data from HDF5 files. For large files (multiple GB and higher), it boasts a much higher performance than the native HDF5-provided command line tools.

CHAPTER 1

Introduction

This project was motivated by both technical and user requirements. First, when using `h5ls` on parallel filesystems with very large data files, I found the performance to be abysmal for even a single application of `h5ls`. Second, when introducing HDF5-formatted output files to engineering analysts used to large ASCII text files, an intuitive way of accessing the output data was needed to lower their barrier to entry. The `h5sh` utility attempts to solve both of these problems by providing an efficient tool to naturally explore arbitrarily large HDF5 data files.

1.1 Installation

The h5sh utility can be installed through python's pip package manager or manually from source.

1.1.1 Stable release

To install h5sh, run this command in your terminal:

```
$ pip install h5sh
```

This is the preferred method to install h5sh, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

1.1.2 From sources

The sources for h5sh can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/sethrj/h5sh
```

Or download the tarball:

```
$ curl  -OL https://github.com/sethrj/h5sh/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

1.2 Usage

This package is meant to be used through the command-line interface (CLI) via the `h5sh` command.

1.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

1.3.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/sethrj/h5sh/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

h5sh could always use more documentation, whether as part of the official h5sh docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/sethrj/h5sh/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

1.3.2 Get Started!

Ready to contribute? Here's how to set up *h5sh* for local development.

1. Fork the *h5sh* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/h5sh.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv h5sh
   $ cd h5sh/
   $ python setup.py develop
   ```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

   ```
   $ flake8 h5sh tests
   $ python setup.py test or py.test
   $ tox
   ```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

7. Submit a pull request through the GitHub website.

1.3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/sethrj/h5sh/pull_requests and make sure that the tests pass for all supported Python versions.

1.3.4 Tips

To run a subset of tests:

```
$ py.test tests.test_h5sh
```

1.3.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 2

Commands

These are the available commands and descriptions inside the h5sh package grouped roughly into categories.

2.1 Navigation

2.1.1 cd

2.1.2 ls

2.1.3 pwd

Print the path to the current HDF5 group.

2.1.4 l

Alias for `ls -l`.

2.1.5 up[p[. . .]]

Shorthand for `cd ..[/..[...]]` to traverse upward in the directory hierarchy. For example, `uppp` is a more typing-friendly equivalent to `cd ../../..`.

2.2 Query

2.2.1 attr

2.2.2 dump

2.3 System

In addition to the system “commands”, you can use `control-C` to clear the current command line and `control-D` to exit h5sh.

2.3.1 exit

Exits the h5sh shell.

2.3.2 help

Lists all available commands.

2.3.3 filename

Print the name of the file being examined.

APPENDIX A

History

A.1 0.1.1 (2019-12-05)

- Fix some dependencies

A.2 0.1.0 (2019-12-05)

- Complete initial capability, documents, and testing.

A.3 0.0.1 (2018-12-07)

- Created cookiecutter package.

APPENDIX B

Authors

B.1 Development Lead

- Seth R Johnson <johnsonsr@ornl.gov>

B.2 Contributors

None yet. Why not be the first?

APPENDIX C

License

APPENDIX D

Indices and tables

- genindex
- modindex
- search