HDF5 in the browser

Native Javascript and WebAssembly tools for working with HDF5

HDF5 User Group Meeting, 2022-05-31

Brian Maranville, NIST Center for Neutron Research

NIST

CHRNS CENTER FOR HIGH RESOLUTION NEUTRON SCATTERING NSF NIST

Why native HDF tools for the web

- …when web-based user interfaces already exist?
  - Cloud-native HSDS
  - @H5Web + h5grove (light Python server)
- These scale to infinity, but less well to zero!

Use case:

New or "casual" users of with a handful of (small) HDF5 files

With a web browser, they can jump in and start working…

jsfive: lightweight Javascript library

- Ported from pyfive https://github.com/jjhelmus/pyfive.git
- Partial implementation of HDF5 file specification in pure ES6
- Loads from JS ArrayBuffer
- Read-only, no modification or create
- Small: bundled ESM including ZLIB = 176kB

https://www.npmjs.com/package/jsfive

https://github.com/usnistgov/jsfive

jsfive: datatype support

- Read Datasets:
  - Float32, Float64
  - Int8, Int16, Int32, Int64
  - Uint8, Uint16, Uint32, Uint64
  - ENUM (as base type)
  - String (fixed length)
  - Returns Array of JS Number type (flat)
- Read attributes:
  - Same numeric types as Datasets, plus VLEN_STRING
  - Returns Object {key: value}

jsfive: filters supported

- Shuffle
- GZIP deflate (from pako library)
- Fletcher32

jsfive: features and limitations

- Tiny ESM source easily integrated with projects
  - babel, create-react-app, rollup, esbuild, etc…
- More recent additions to the HDF5 file spec not implemented
  - e.g. datatype message, only version 1
  - Files written with newer features may fail to fully read
  - It will read what it can, though – without panic
- No ARRAY or COMPOUND datatype support
- No partial dataset retrieval (slicing)

h5wasm: full-featured wasm library

- Created to support more HDF5 features than jsfive
- WASM component written in C++, against libhdf5 C API
- JS component written in TypeScript
- Leverages emscripten virtual file system
- Read, Write or Append
- Bundled ESM including ZLIB = 3.2MB

https://www.npmjs.com/package/h5wasm

https://github.com/usnistgov/h5wasm

h5wasm: datatype support

- Reads any Dataset (or Attribute value)
  - Converts simple datatypes to nearest JS native type, e.g.
    - 4-byte float to JS Float32Array
    - 8-byte unsigned int to BigUint64Array
    - ENUM datatypes to base types (member names in metadata)
    - COMPOUND, ARRAY datatypes to nested JS Array(s) of base types
  - If datatype is not recognized, returns raw Uint8Array bytes
- Writes simple datatypes automatically
  - All TypedArray variants
  - String as VLEN_STRING
- Can manually specify dtype and write Uint8Array of bytes

h5wasm: filters supported

- gzip
- Shuffle
- Fletcher32

Adding more filters:

- Could be compiled for "dynamic linking"
  - MAIN_MODULE and SIDE_MODULE mode
- Or compile and static link

h5wasm: additional features

- Slicing retrieves partial datasets (using Dataspace):
  - Slice notation similar to python:

```
[
  [axis0_start, axis0_end],
  [axis1_start, axis1_end], …
]
```

- "boolean" datasets as written by h5py are read into boolean arrays (if datatype is ENUM {FALSE:0, TRUE:1})
- "to_array" method returns nested JS Array of values matching shape

h5wasm in current use

- Data previewers at NIST Center for Neutron Research
  - HDF5 general viewer
  - VSANS viewer
  - CANDOR viewer

- Integration with PANOSC H5web NeXus viewer
  - https://www.npmjs.com/package/@h5web/h5wasm
  - https://h5web.panosc.edu/h5wasm

Future developments of h5wasm

- Rewrite to C++ HDF5 API? (from C API)
- Add more support for data writing (Compound)
- Support partial file loading over https (lazyFile)
  - Requires WebWorker + Range requests on server
  - Proof of principle already done

...coming soon: pyodide + h5py

- In current master branch, pyodide includes h5py
  - Based on HDF5 v1.13.1
  - Not planned for next release
- Allows loading and working with HDF5 from
  - JupyterLite
  - https://observablehq.com

Acknowledgments

- Thanks to Loïc Huder and other H5Web developers for collaboration on h5wasm features
- Support was provided by the Center for High Resolution Neutron Scattering, a partnership between the National Institute of Standards and Technology and the National Science Foundation under Agreement No. DMR-2010792.