NetCDF4 / HDF5 Format Extension Pack (NFEP): Transparently Adding More Formats to NetCDF4 and HDF5

# Identification and Significance of the Proposed Innovation

### Innovation Overview

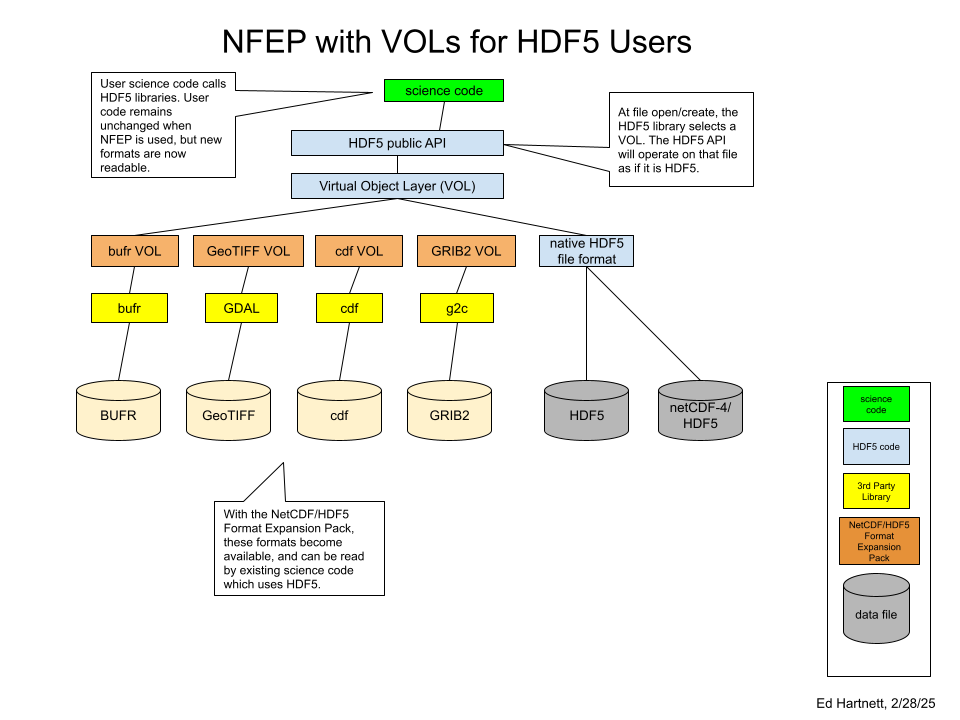
The NetCDF4/HDF5 Format Extension Pack (NFEP) allows netCDF4 and HDF5 users to read all common Earth science data formats, transparently. The extension pack will add read-only capability for other common Earth science formats: CDF, BUFR, GeoTIFF, and GRIB2. Existing netCDF4 and HDF5 code, including Fortran and Python codes, will be able to read data files of these formats without any modifications to the source code.

NFEP takes advantage of the Virtual Object Layer (VOL) architecture of HDF5, and the ability of netCDF4 to use new HDF5 features. The NFEP uses the VOL architecture to provide plugins to work with additional data formats. Rather than requiring scientists to convert between disparate formats—often losing information in the process—NFEP allows specialized data representations to be accessed through a unified interface while preserving their unique characteristics and optimizations. This innovation addresses three critical challenges faced by NASA Earth Science data systems:

1. **Data Format Proliferation**: Earth Science research spans multiple domains, each with specialized formats optimized for particular data types. Currently, interoperability between these formats requires complex, often lossy conversions.
2. **Computational Efficiency**: Domain-specific formats like GRIB and BUFR implement specialized compression and representation techniques that significantly reduce storage requirements and processing time for specific data types.
3. **Metadata Preservation**: Specialized formats contain rich, domain-specific metadata that is often lost when converting to general-purpose formats like standard netCDF.

The NFEP innovation creates a unified framework where the benefits of specialized formats can be retained while simultaneously enabling interoperability with the broader scientific data ecosystem through netCDF4 and HDF5.

Architecture of our solution for HDF5 and netCDF4 applications is shown on Figure 1 and Figure 2 below.



*Figure 1: The addition of the NetCDF4/HDF5 Format Expansion Pack allows HDF5 users to access data in BUFR, GeoTIFF, CDF and GRIB2 formats (read-only). This is accomplished using the native libraries which handle these formats.*



*Figure 2: For netCDF4 users, the NetCDF4/HDF5 Format Expansion Pack adds the formats available through the existing HDF5 interface. Since these formats are automatically detected by HDF5 at read time, the correct VOL connector can be used, transparently to the users.*

### Relevance to NASA Needs

The proposed innovation directly addresses needs identified in Subtopic S17.01: Technologies for Large-Scale Numerical Simulation, specifically the call for "technologies that can enhance data processing and interoperability across heterogeneous computing resources."

NASA's Earth Science missions generate unprecedented volumes of observational data in diverse formats. These datasets must be integrated with simulation outputs and analyzed across heterogeneous computing environments, including exascale environments. Current approaches require scientists to either:

1. Convert all data to a common format, losing specialized optimizations and often metadata;
2. Maintain separate processing pipelines for each format, increasing complexity and reducing interoperability; or
3. Develop custom, one-off solutions for specific format pairs, creating maintenance challenges.

NFEP addresses these challenges by enabling seamless integration of specialized formats within NASA's existing netCDF4 and HDF5 codes and infrastructure. This innovation is particularly relevant to NASA's needs in:

* **Increase the achievable scale and complexity of computation, data ingest, and/or data assimilation required with large-scale numerical simulations**: NASA researchers use petabytes of data across multiple formats. NFEP will reduce the need for format conversion, preserving data fidelity while improving access of these data with existing exascale science codes.
* **Enhance the efficiency and effectiveness of NASA's supercomputing operations and services**: By maintaining the optimized storage characteristics of specialized formats like GRIB and BUFR while providing netCDF4 access patterns, NFEP will reduce storage requirements and improve processing performance. Programmers will no longer have to wrangle diverse data formats into one, or copy existing data in new formats. Existing data of diverse formats will become readable to existing netCDF4 and HDF5 science codes, including exascale codes on HPC systems.

NASA's 2023 Earth Science Technology Strategy specifically highlights data interoperability as a critical need for advancing Earth system science. NFEP directly supports this strategic direction by enabling more efficient data integration across sensors, models, and domains.

### Innovation Relative to State of the Art

Current state-of-the-art approaches to scientific data format interoperability include:

**1. Format Conversion Tools**Tools such as GDAL/OGR and wgrib provide conversion between formats but typically lose format-specific optimizations and often metadata. These conversions create duplicate data and introduce version control challenges.

**2. Format-Specific Libraries** Libraries such as BUFR, G2C, G2, and CFGRIB provide high-performance access to specific formats but require application developers to implement multiple interfaces and manage format detection.

**3. General Data Models with Format-Specific Implementations** The Common Data Model (CDM) approach used by libraries like XARRAY provides a unified data model but often sacrifices performance and specialized capabilities of native formats.

**4. Web-Based Data Services** Systems like THREDDS and OPeNDAP provide format-independent access but typically require server-side format conversion and don't preserve format-specific features.

Our NFEP innovation advances beyond these approaches in several key ways:

**1. Format Preservation with Unified Access** Unlike conversion tools, NFEP preserves the native format while providing standardized access. Data remains in its optimized form while appearing as a standard netCDF dataset to applications.

**2. Improve Programmer Productivity** NFEP allows programmers to access datasets which would otherwise require custom programming. Since data are transparently available to netCDF4 and HDF5 APIs, existing science codes, including exascale codes, would gain (read-only) access to these formats, without extra code or copying the data.

**3. VOL Plugin Architecture for Format Extensions** NFEP uses an existing plugin architecture for format extensions that standardizes how new formats are integrated and documented, rather than requiring one-off integrations. Further format adaptors may be added, and adaptors may also be made read/write, in future releases (beyond the scope of this proposal.)

The table below summarizes the advantages of NFEP compared to current state-of-the-art approaches.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Capability** | **Format Conversion** | **Format-Specific Libraries** | **General Data Models** | **Web Services** | **NFEP** |
| Preserve format optimizations | ❌ | ✅ | ❌ | ❌ | ✅ |
| Single API for all formats | ❌ | ❌ | ✅ | ✅ | ✅ |
| Full metadata preservation | ❌ | ✅ | ❌ | ❌ | ✅ |
| No duplicate storage | ❌ | ✅ | ❌ | ❌ | ✅ |
| Remote data access | ❌ | ❌ | ❌ | ✅ | ✅ |
| Format extension registry | ❌ | ❌ | ❌ | ❌ | ✅ |

In summary, NFEP represents a significant advancement beyond current approaches by combining the preservation of format-specific optimizations with standardized access patterns, and compatibility with existing netCDF4 and HDF5 codes. This innovative approach will dramatically improve NASA's ability to efficiently manage and analyze diverse Earth Science datasets while reducing storage requirements and computational overhead. It will allow NASA programmers to use existing science codes on GRIB2, BUFR, CDF, and GeoTIFF datasets.

# Technical Objectives

### Phase I Objectives:

1. Design the architecture for the NetCDF4 Format Extension Pack.
2. Develop format extensions for CDF, GeoTIFF, GRIB, and BUFR, leveraging existing libraries.
3. Demonstrate interoperability with existing netCDF4 tools and NASA data systems.
4. Evaluate performance impacts and optimization opportunities.

**Deliverables**

At the end of Phase I efforts, version 1.0 of the NFEP will be released, providing netCDF4 and HDF5 programs with read-only access to GRIB, BUFR, GeoTIFF, and CDF files.

The VOL format adaptors must map the native format to the HDF5 model. That is, they must provide the kind of multi-dimensional arrays, and attribute information, which the VOL connector can serve up to HDF5. In this way, HDF5 users will see the data as if it were stored in HDF5. Similarly, netCDF4 users will see the data as if it were stored in netCDF4.

Risks include difficulty mapping each of the target formats to the necessary HDF5 abstraction. These risks are mitigated by:

* The generality of the HDF5 data model.
* The chosen data formats all store geophysical data, which can be expressed as multi-dimensional arrays with attributes.

When mapping the data to arrays and attributes, attributes will be chosen which match those used in the CF Conventions, as much as possible. Full CF compatibility would be a useful further goal, but is outside the scope of this proposal.

# Work plan

**Detailed Task Description**

Tasks include:

1. Creation of NFEP framework, build system, and testing and continuous integration system, setting up the base for the NetCDF4/HDF5 Format Extension Pack (NFEP). This includes:
   * 1. Establishing a GitHub repository for version control and team collaboration.
     2. Developing a CMake-based build system to manage the compilation process across different platforms.
     3. Creating documentation build processes.
     4. Implementing unit test builds to ensure code quality.
     5. Establishing a continuous integration system to automate testing and integration of code changes.
2. Creation and testing of VOL plugins Developing and testing Virtual Object Layer (VOL) plugins for various data formats:
   * 1. GRIB2 reader: Utilizing the NOAA GRIB2 C library (NCEPLIBS-g2c) to enable reading GRIB2 files.
     2. BUFR reader: Utilizing the NOAA BUFR library (NCEPLIBS-bufr) to enable reading BUFR files.
     3. GeoTIFF reader: Using the GDAL library and code from the OPeNDAP project to map to netCDF files.
     4. CDF reader: Using the NASA CDF library to read CDF data as NetCDF.
3. Release of v1.0 Releasing the NetCDF4 Format Expansion Pack, version 1.0, which will provide netCDF4 programs with read-only access to GRIB, BUFR, GeoTIFF, and CDF files.

**NFEP Framework**

The basic framework for NFEP consists of a repository to manage the code, managed using agile principles. Creating the framework and agile development system includes:

1. CMake build system
2. Unit tests
3. Continuous Integration system
4. Documentation
5. Spack package file to integrate cleanly with Spack.

**HDF5 VOL Plugins**

The HDF5 Virtual Object Layer (VOL) is an abstraction layer in the HDF5 library that intercepts API calls to access objects in an HDF5 container and forwards those calls to a VOL connector, which implements the storage.

Key aspects of the VOL architecture include:

**Location within the HDF5 library:** The VOL is situated just under the public API. Storage-oriented public API calls trigger the library to perform sanity checks and then invoke a VOL callback. This callback resolves to an implementation within the VOL connector selected when the file was opened or created.

**Functionality:** The VOL connector carries out necessary operations before control is returned to the library for any final operations, such as assigning IDs for newly created datasets. The VOL connector handles most of the functionality for calls that utilize the VOL.

**Limited Caching:** Most of the HDF5 caching layers (metadata and chunk caches, page buffering) are not available to external connectors because they are implemented in the HDF5 native VOL connector and cannot be easily reused.

**Selective API calls:** Not all public HDF5 API calls pass through the VOL. The VOL is used only by calls that require manipulating storage. Calls related to dataspace, property list, or error stack do not use the VOL.

**Variable Implementation:** Not every VOL connector implements the complete HDF5 public API. Certain features, such as variable-length types, may not be developed or have equivalents in the target storage system. Many HDF5 public API calls are specific to the native HDF5 file format and may not be useful in other VOL connectors.

**Implementation Methods:** A VOL connector can be implemented as a shared or static library linked to an application, as a dynamically loaded plugin (shared library), or as an internal connector built into the HDF5 library.

**Native File Format Connector:** The native file format connector handles native HDF5 (\*.h5/hdf5) files. It is an internal VOL connector and a core part of the HDF5 library. It cannot be unloaded and is always present.

**V1.0 Release**

The V1.0 release will include VOL connectors for the newly supported formats. Also included will be documentation, unit testing, and a Spack package file, to allow Spack to easily build and install the NFEP.

**Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Month** | **Hours** | **Notes** |
| NFEP framework | 1 | 170 | Includes GitHub repository, CMake-based build system, documentation build, unit test build, and continuous integration system. |
| GRIB2 plugin | 2 | 170 | Uses the NOAA GRIB2 C library (NCEPLIBS-g2c) to read GRIB2 files. Unit tests and documentation. |
| BUFR plugin | 3 | 170 | Uses the NOAA BUFR library (NCEPLIBS-bufr) to read BUFR files. Unit tests and documentation. |
| GeoTIFF plugin | 4 | 170 | Uses the GDAL library and code from the OPeNDAP project to map to netCDF files. Unit tests and documentation. |
| CDF reader | 5 | 170 | Uses the NASA CDF library to read CDF data as NetCDF. Unit tests and documentation. |
| Release of v1.0 | 6 | 170 | Release of the NetCDF Format Expansion Pack, version 1.0. Unit tests and documentation. |

**Resource Allocations including Estimated Task Hours**

|  |  |  |
| --- | --- | --- |
| Resource | Allocation | Estimated Task Hours |
| PI - Edward Hartnett | 100% | 1020 hours |
| Elena Pourmal - QA, outreach | 20% | 204 hours |

**Planned Accomplishments**

The project will make the following releases:

|  |  |  |
| --- | --- | --- |
| **Release** | **Project Month** | **Notes** |
| v0.1 | 2 | NFEP framework, build system, and GRIB2 reader. |
| v0.2 | 3 | Add BUFR reader. |
| v0.3 | 4 | Add GeoTIFF reader. |
| v0.4 | 5 | Add CDF reader. |
| v1.0 | 6 | Full release. |

All work will be done by the PI in Lifeboat, LLC offices. All software will be developed and maintained in accordance with agile software practices, including full unit testing, continuous integration, and regular releases with full documentation.

# Related R/R&D

The team has used the plugin architecture in several released products, both of which enjoy significant use:

* The netCDF-C HDF4 read-only layer.
* The integration of the ParallelIO package with netCDF-C.
* HDF5 1.14.\* with VOL architecture and HDF5 VOL plugins

**NetCDF Read-Only HDF4 SD Layer**

Since the 4.2.0 release (2012) the netCDF-C library has included an optional reader for HDF4 SD files. This reader code consists of about 1000 lines of C code in the libhdf4 subdirectory of the netCDF-C code.

The code in this subdirectory implements the functions needed for HDF4 SD files to be read and understood, and the data served up through the netCDF-C API as if it were netCDF data. Since the netcdf-fortran library uses the netCDF-C library, this makes HDF4 SD data available to all netCDF Fortran codes, including codes on exascale systems.

The code in libhdf4 provides a complete mapping of all HDF4 SD file data types, attributes, and shapes, to the netCDF data model. The result is that Fortran programs written with netcdf-fortran can read HDF4 SD data, as if it were netCDF data.

NASA has many HDF4 datasets including the Moderate Resolution Imaging Spectroradiometer (MODIS) imagery from the Terra and Aqua satellites. Making these datasets transparently available to NASA codes, including exascale codes on HPC systems, allows models and other large codes to be extended more efficiently, with less I/O programming.

The successful integration of HDF4 SD with netCDF-C since 2012 has allowed NASA programmers to read important datasets without copying them into new formats. NASA has significant HDF4 data holdings.

**Integration of NetCDF with PIO**

The ParallelIO package is a library which provides sophisticated functionality for reading and writing netCDF and HDF5 files on HPC systems, including exascale systems. Using PIO, existing netCDF code, including netcdf-fortran code, can be used in code which is scaled up to tens or hundreds of thousands of processors. The ParallelIO library manages the I/O burden from all processors to the more limited external storage hardware by buffering and aggregating the data efficiently and transparently.

This integration with netCDF is possible because of the netCDF-C dispatch layer. As with the HDF4 SD reader, the dispatch layer is used to call the ParallelIO functions when necessary to read or write data. This allows ParallelIO to take control of the I/O on HPC systems, and provide efficient and fast I/O.

This successful integration demonstrates that significant Fortran code reuse can be achieved transparently.

**HDF5 1.14.0 and HDF5 VOL connectors**

Stable HDF5 VOL architecture and VOL layer APIs were released in HDF5 1.14.0 and are actively maintained by The HDF Group. Conceptual high-level HDF5 architectural diagram is presented on Figure 3. An HDF5 application uses HDF5 APIs to describe the application’s data in terms of HDF5 objects (file, group, dataset, attribute, etc.) and stores the data on a specified storage device, e.g., parallel file system, node local storage, or object store. HDF5 I/O is performed by the Virtual Driver File (VFD) layer and its drivers, or by special VOL connectors.

The VFD layer is an abstraction layer at the bottom of the HDF5 library, which presents the underlying storage system as an extensible vector of bytes (i.e., binary HDF5 file). Historically, this abstraction layer has been used to allow the HDF5 library to run on different operating systems with different file I/O APIs, to simulate large files on file systems with a 2 GB max file size, access data on S3 store, and enable scaling for parallel HDF5 applications.

The VOL layer can be thought of as an abstraction layer just below the HDF5 APIs. It provides interfaces to the VOL connectors that use the HDF5 API and some utilities provided by the library to implement data storage in an arbitrary format and on arbitrary devices while supporting the HDF5 API and data model.

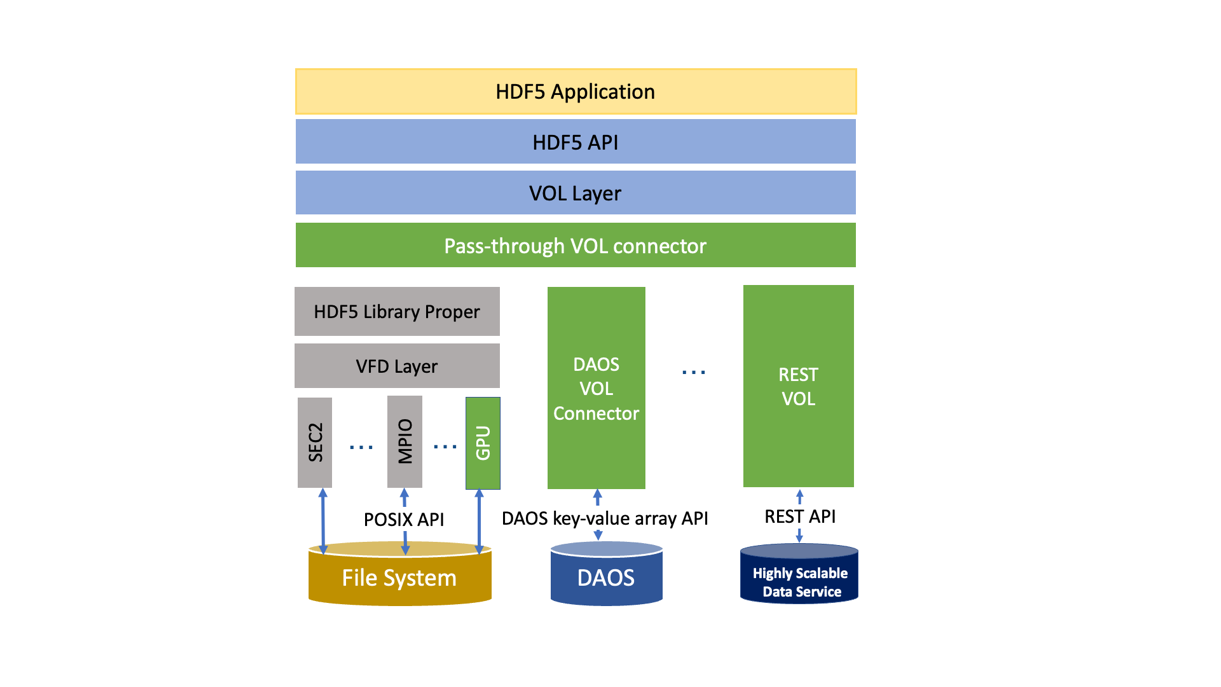


Figure 3: HDF5 architecture. VOL layer redirects HDF5 calls to the HDF5 library proper to perform POSIX I/O via Virtual File Driver (VFD) layer resulting in binary HDF5 file and to VOL connectors that map HDF5 objects to storage and storage specific I/O API.

Figure 3 shows pluggable DAOS VOL and REST VOL connectors that use connector specific API for performing I/O to DAOS storage and passing I/O requests to the Highly Scalable Data Service that in turns talks to sharded HDF5 data in S3. The VOL layer also makes possible “pass-through” VOL connectors that call other VOL connectors. The VOL connectors are registered with the HDF5 library using connector’s identification numbers that are assigned and maintained by The HDF Group. The list of the registered VOL connectors can be found at

<https://support.hdfgroup.org/documentation/hdf5-docs/registered_vol_connectors.html>. All VOL connectors except DAOS use HDF5 storage formats. Our work will leverage VOL’s capabilities to access data in non-HDF5 formats.

# Key Personnel and Bibliography of Directly Related Work

**Principle Investigator - Edward Hartnett**

Edward Hartnett is the primary author of NetCDF-4, a widely used, freely available software library for scientific data relied upon by NASA, NOAA, the European Space Agency (ESA), and climate and meteorology researchers worldwide. Originally funded by NASA, and later supported by NSF through UCAR, NetCDF has become the standard data format for Earth science data, enabling efficient storage, retrieval, and sharing of critical scientific datasets. It is a core component of NASA and NOAA’s GOES satellite data processing systems, helping ensure that satellite observations are properly stored, distributed, and accessible to researchers, forecasters, and decision-makers.

Edward continues to play an active role in the ongoing development and optimization of NetCDF, regularly contributing performance improvements, bug fixes, and new features to keep pace with evolving scientific and operational requirements. In 2022, he led a major enhancement to NetCDF’s data compression capabilities, dramatically improving storage efficiency for large-scale climate and satellite datasets.

In addition to his work on NetCDF, Edward is a co-author and active contributor to the Parallel IO (PIO) library, a high-performance C/Fortran library designed to optimize parallel I/O operations using the MPI (Message Passing Interface) library. PIO is a key component of NCAR’s Community Earth System Model (CESM) and is used in other major weather and climate models to efficiently handle vast amounts of environmental simulation data. Designed to run on some of the world's most powerful HPC systems, PIO enables scalable, high-performance data processing critical for advanced climate modeling, numerical weather prediction, and scientific research.

Edward has also led and mentored software engineering teams, including serving as supervisor of the Production Software Team within Ground Data Systems for NASA missions such as Messenger, Cassini, MAVEN, MMS, SORCE, AIM, and TSIS. In this role, he ensured the successful delivery of mission-critical data processing software, established Agile development processes, and implemented comprehensive code reviews to uphold software quality for operational NASA missions.

Currently, Edward works at NOAA’s Environmental Modeling Center (EMC), where he contributes to the development and maintenance of NCEPLIBS, a key suite of libraries supporting the Unified Forecast System (UFS) and other numerical weather prediction (NWP) models. His work focuses on improving performance, scalability, and scientific accuracy for some of the world’s most complex environmental modeling systems. In addition, Edward actively explores AI-based solutions to streamline development processes, improve code quality, and accelerate scientific software development, applying cutting-edge techniques to enhance NOAA’s modeling and forecasting capabilities.

**Bibliography**

Edward Hartnett, Hang Lei, Alexander J.W. Richert, Alyson Stahl, “A New API for NOAA's GRIB2 Libraries” Poster presented at the AGU Fall Meeting 2024.

Alexander J.W. Richert, Edward Hartnett, Hang Lei, Jeff Ator, George Gayno, Alyson Stahl, “Modernizing scientific code development and testing: NCEPLIBS perspectives on continuous integration” Poster presented at the AGU Fall Meeting 2024.

Edward Hartnett, Jeff Ator, Hang Lei, AlexanderJ.W. Richert, Jack Woollen, Alyson Stahl, George Vandenberghe, “NCEPLIBS Libraries Modernization” Presented at the SENA Summit, Boulder, Colorado, September, 2024

Edward Hartnett, Jeff Ator, Hang Lei, Alexander J.W. Richert, Jack Woollen, “NCEPLIBS GRIB and BUFR Libraries: Maintaining and Modernizing NOAA's Libraries for WMO Data Formats” Poster presented at the AGU Fall Meeting 2023 (see extended abstract).

Edward Hartnett, Jeff Ator, Hang Lei, Alexander J.W. Richert, Jack Woollen, “Benchmarking Zstandard and Quantize Compression in NetCDF” Poster presented at the AGU Fall Meeting 2023.

Edward Hartnett, “NetCDF Compression Improvements” Presented at the SENA Summit, Boulder, Colorado, June, 2023

Hartnett, Edward. “Better Compression for UFS with Support from the NetCDF Community.” Presentation delivered at the Unifying Innovations in Forecasting Capabilities Workshop, July 2023.

Hartnett, Edward. “Unit Testing NCEPLIBS.” Poster presented at the Workshop on Correctness and Reproducibility for Climate and Weather, November 9, 2023, Boulder, CO.

Edward Hartnett, Charlie Zender, “Advanced Compression Algorithms for netCDF Datasets with the Community Codec Repository” Presented at NSF CYBERINFRASTRUCTURE FOR SUSTAINED SCIENTIFIC INNOVATION (CSSI) PRINCIPAL INVESTIGATOR MEETING, July, 2022.

Edward Hartnett, Hang Lei, Brian Curtis, Kyle Gerheiser, “Improving Documentation, Testing, Process, and Code for Legacy NOAA GRIB2 C Fortran Libraries” Presented at the Improving Scientific Software Conference, NCAR, Boulder, Colorado, April, 2022 (see video).

Edward Hartnett, Charlie Zender, “Adding Quantization to the NetCDF C and Fortran Libraries to Enable Lossy Compression” Presented at the EGU Meeting, Vienna, Austria, April, 2022.

Edward Hartnett, Charles S. Zender, Ward Fisher, Dennis Heimbigner, Hang Lei, Brian Curtis, Kyle Gerheiser, “Quantization and Next-Generation Zlib Compression for Fully Backward-Compatible, Faster, and More Effective Data Compression in NetCDF Files” Presented at the AGU Fall Meeting 2021 (see extended abstract).

Edward Hartnett, Charles S. Zender, “ADDITIONAL NETCDF COMPRESSION OPTIONS WITH THE COMMUNITY CODEC REPOSITORY” Presented at the AMS Meeting 2021.

NETCDF-4 PERFORMANCE IMPROVEMENTS OPENING COMPLEX DATA FILES, 35th Conference on Environmental Information Processing Technologies, AMS Annual Meeting · Jan 10, 2019

Recent Development in NetCDF Libraries and FormatsRecent Development in NetCDF Libraries and Formats, European Geophysical Union 2001 Poster: EGU2011-8288 · Apr 1, 2011

Rew K Russell, Edward Hartnett, John L. Caron, NetCDF-4: Software Implementing an Enhanced Data Model for the Geosciences, January 2006, Conference: 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology

**Elena Pourmal, PM**, **Software developer,** has over 20 years of experience as an HDF5 developer and project manager. During her time at The HDF Group, she has focused on managing software maintenance and development of HDF5 features introduced in version 1.10.0 and later releases, including SWMR, VDS, parallel compression, and performance improvements, as well as the HDF5 DAOS VOL connector. Currently, she works on the HDF5 multithreaded VOL connector and support for sparse data in HDF5.

Ms. Pourmal has collaborated with multiple organizations on HDF5-based projects, including HDFEOS5, netCDF-4, NeXus, and the binary STEP standards. She has also taught numerous tutorials on HDF5 and organized HDF5 BoFs at SC conferences and HDF User Group meetings. In addition to her technical work, she performs marketing activities and engages with potential Lifeboat, LLC customers. On this project, Ms. Pourmal will serve as project manager and software developer focusing on data modeling, design and code reviews, testing, documentation, benchmarking, and outreach.

**Bibliography**

Soumagne J, Henderson J, Chaarawi M, Fortner N, Breitenfeld S, Lu S, Robinson D, Pourmal E, Lombardi J. Accelerating HDF5 I/O for Exascale using DAOS. IEEE Transactions on Parallel and Distributed Systems. 2022 April; (4):903. Available from: https://ieeexplore.ieee.org/document/9490299 DOI: 10.1109/TPDS.2021.3097884

Elena Pourmal (2024, August 13). Upcoming New HDF5 Features: Progress on Multi-thread, Sparse Data Storage, and Encryption in HDF5. 2024 HDF5 User Group Meeting (HUG24), Illinois Institute of Technology, Chicago, IL. Zenodo. <https://doi.org/10.5281/zenodo.13308713>

# Market Opportunity

Lifeboat, LLC is a startup and a spin-out of The HDF Group (THG), a nonprofit organization that maintains the free and open-source HDF5 software (<https://github.com/HDFGroup/hdf5>). Founded in August 2021, Lifeboat was established to commercialize H5+, a suite of technologies designed to address the limitations of HDF5 and facilitate its broader integration with commercial products.

The initial versions of H5+ will include open-source, pluggable I/O drivers, storage connectors, and command-line tools that seamlessly integrate with the existing HDF5 software. Over time, H5+ will evolve into a modular, multithreaded HDF5 library equipped with a comprehensive set of tools and I/O plugins, enabling access to HDF5 data across diverse storage systems, including traditional file systems, object stores, and cloud environments. Additionally, Lifeboat provides consulting, support services, and development expertise for creating HDF5 plugins.

We see our market opportunities in two areas:

### Addressing the Growing Demand for NASA Data in Private Sector

In 2023, NASA’s Distributed Active Archive Centers (DAACs) accumulated over 100 petabytes (PB) of remote sensing data[[1]](#footnote-1). This volume is expected to grow exponentially with upcoming missions like the NASA-ISRO SAR Mission (NISAR). Remote sensing data is invaluable to the private sector, driving informed decision-making across industries such as agriculture, urban planning, environmental monitoring, and disaster management. High-resolution Earth imagery enables precise analysis, predictive modeling, and improved risk mitigation strategies, leading to greater efficiency and cost savings. For example, geospatial finance integrates geospatial data into financial theory and practice. It uses satellite imagery to accurately map and monitor the Earth's resources, ecosystems, and climate. “Such imagery provides valuable data and insights into the environmental impact of investment activities, assisting investors in identifying potential risks and opportunities, and tracking the effectiveness of sustainable finance initiatives over time” [1]. NASA’s imagery becomes immediately available to the financial applications that use h5py (Python interface to HDF5) and VOL connectors to NASA’s data formats.

A unified data access interface (HDF5 or netCDF4) significantly reduces the cost of application development and maintenance. By enabling seamless access to NASA’s data across various formats[[2]](#footnote-2) , our solution eliminates the need for costly application updates when new or proprietary formats emerge. Organizations can leverage our technology without substantial investments in software development, and by removing the need for data conversion during ingestion, we dramatically accelerate data processing and research workflows.

### 

### Simplifying AI/ML Integration

Artificial intelligence and machine learning (AI/ML) play an increasingly vital role in Earth science, climate research, and weather forecasting, particularly in the private sector. Our data ingestion approach simplifies the use of NASA’s datasets within AI/ML frameworks such as PyTorch and TensorFlow. These frameworks rely on the h5py Python interface to interact with the HDF5 library and VOL connectors, ensuring seamless data access for advanced analytics and model training.

### Overcoming Challenges in Adoption

While there are no technical barriers to implementing our solution, transitioning from research and development to widespread adoption requires overcoming key challenges:

**Limited awareness among end users** – While many users of HDF5 and netCDF4 are familiar with the technology, they may not be aware of its advanced capabilities for accessing data beyond traditional HDF5 file formats. To address this, we will actively market our approach through collaborations with application developers, private companies, and by engaging with the community via publications, conferences, and workshops.

**Inadequate feedback mechanisms between developers and users** – To ensure a smooth user experience, we will establish a structured feedback system for VOL connectors, providing direct support and gathering user input. Lifeboat has a long history of engagement with the HDF5 and netCDF4 user communities. Additionally, we are utilizing TABA funds from other projects to develop a web portal, where users can submit questions, report issues, and access training materials.

Ultimately, the success of our innovation will depend on its ease of use, strong outreach efforts, and exceptional customer support. By addressing these challenges, we aim to drive widespread adoption and deliver tangible benefits to organizations leveraging NASA’s vast data resources.

# Facilities and Equipment

Lifeboat, LLC occupies 2,500 square feet of office space in Champaign, Illinois and Laramie, WY. Each member of the technical team has Linux and/or macOS development systems and will have access to EC2 instances with different versions of Linux and C compilers and S3 storage. We estimate the cost of EC2 instances required for development is less than $5000 per year. The team will utilize GitHub capabilities to perform CI testing and releases on a variety of systems, including Windows and macOS.

# Subcontractors and Consultants

# N/A

# Related, Essentially Equivalent, and Duplicate Proposals and Awards

N/A

1. <https://www.earthdata.nasa.gov/about/program-highlights/2023> [↑](#footnote-ref-1)
2. <https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/data-formats> [↑](#footnote-ref-2)