Xarray: N-D labeled arrays and datasets in Python

Proposal to the Chan Zuckerberg Initiative Essential Open Source Software for Science Program

Proposal Team

1. Joe Hamman, National Center for Atmospheric Research
2. Ryan Abernathey, Columbia University, Lamont Doherty Earth Observatory
3. Benoît Bovy, Freelance Research Software Engineer
4. Stephan Hoyer, Google Research
5. Deepak Cherian, National Center for Atmospheric Research

Proposal Details

6. **Proposal Title:** Xarray: N-D labeled arrays and datasets in Python
7. **Did you previously apply for funding for this or a related proposal under the EOSS program?:** No
8. **Proposal Purpose (required):**
   Limit to one sentence (maximum of 255 characters, including spaces)

To grow the use of Xarray in the biosciences as a foundational data model and computational toolkit for multidimensional labeled arrays.

9. **Abstract / Proposal Summary (required):**

   Provide a short summary of the application (maximum of 250 words)

Xarray is an open source Python project that makes working with labelled multi-dimensional arrays elegant, intuitive, and efficient. Real-world datasets are more than raw numbers; they

have labels which describe how array values map to locations in dimensions such as space and time; metadata that describes how the data was collected and processed etc; and are often a collection of multiple fields on a common grid.

Xarray embraces this complexity and provides tools for users to easily analyze, manipulate and visualize data using labels as well as preserve important metadata attributes. Xarray combines an expressive API inspired by Pandas with Unidata's Common Data Model for self-described scientific data.

Originally developed for analyzing multi-dimensional climate and weather data, the reach of Xarray now extends across a broad swathe of scientific domains. Xarray is particularly well suited for analyzing the large and heterogeneous biomedical datasets produced by a wide range of experimental research tools. Existing applications of Xarray in the biomedical sciences are found from bioimaging to single-cell genomics. Below, we highlight a short list of high-impact applications that make use of Xarray:

- Image based transcriptomics (Starfish)
- Neuroimaging (Allen SDK)
- Single cell RNA-seq data (Zaini et al., 2018)
- Embryo phenomics (Tills et al., 2018)

This proposal targets specific development goals that will make it easier for new scientific communities to use and extend Xarray. Our focus is on community engagement, documentation and three areas of technical development: flexible grids and indexes; flexible arrays and computing backends; and flexible storage backends.

10. **Workplan (required):**

    Provide a description of the proposed work the applicants are requesting funding for, including resources the applicants will provide that are not part of the requested funding. For software development related work (e.g., engineering, product design, user research), specify how the work fits into the existing software project roadmap. For community outreach related activities (e.g., sprints, training), specify how these activities will be organized, the target audience, and expected outcomes (maximum of 750 words)

Xarray's current challenge is to continue providing its core labeled multi-dimensional array functionality while enabling easy extensibility for custom scientific applications. The user community has voiced a number of specific needs in this regard:

- Non-regular grids (e.g., staggered and unstructured meshes),
- Lazily computed indexes (e.g., for coordinate systems),
- Custom arrays (e.g. GPU arrays, sparse arrays, unit aware arrays), and
- New file-formats.

Xarray may not solve all of these issues directly, but these areas motivate Xarray to provide more extensible interfaces. Our proposal addresses these challenges by prioritizing key parts of our roadmap that are *unlikely to proceed without external funding*. Below, we detail four distinct development initiatives:

1. *Community Engagement:*
   Xarray is already equipped to be a foundational tool for a diverse set of domain applications. We believe that its primary shortcoming is not technical, but rather social — and that we should focus effort on engagement with new communities.
   A. Our **online documentation** is skewed toward geoscientists familiar with the Pandas API and the concepts behind NetCDF. Enabling other communities to access Xarray's capabilities requires a substantial overhaul of our documentation, with a focus on accessibility to new users. Specific goals include:
      - A dask.org-like **splash page** with catchy images, and short example code demonstrating common idioms.
      - Expanded set of domain-specific interactive tutorials that run on Binder and walk through core concepts and features.
   B. Conducting two user surveys to better understand the existing and potential user communities, and to guide future development efforts.

2. *Flexible Grids and Indexes:*
   One of Xarray's core capabilities is the easy extraction of data from Xarray objects using coordinate labels instead of array indices. The Pandas-inspired Xarray syntax `` `temperature.sel(place="Boston")` `` is more intuitive and less error-prone compared to NumPy syntax: `` `temperature[0]` ``.

   This capability is supported by the underlying concept of "indexes", currently implemented using `pandas.Index` classes. While our current implementation has been sufficient for early adopters with highly structured data, we now have frequent reports of users wanting to use Xarray to extract and analyze less-structured datasets. To address these indexing challenges, we propose to:

   - Refactor Xarray's indexes to position them as first-class members of the Xarray data model,
   - Create an external interface for users to plug in custom indexes to Xarray objects
   - Wrap existing indexes within the new model and provide other indexes for common uses (e.g., `KDTreeIndex` for spatial indexing)

3. *Flexible Storage Backends:*

   Xarray provides a suite of integrated storage backends that allow users to easily read and write data from a wide selection of data formats (e.g. HDF5/NetCDF, Zarr, Pickle,

etc). Over time, our backends module has grown organically in size and complexity resulting in a fragile internal API that is difficult for even experienced Xarray developers to use.

Rather than continuing to maintain storage backends within Xarray itself, we plan to provide third-party backend developers with an abstract backend class and a plugin API. This will let developers create and distribute compatible backends without the need for explicit xarray support.

Specific development objectives include:

- Exposing an abstract backend for writing new storage systems; API for common backend utilities (GH1970),
- Plugin interface for third party backends (GH1970), and
- Breaking out infrequently used backends to third-party packages.

11. **Milestones and Deliverables (required):**

List expected milestones and deliverables, and their expected timeline. Be specific and include (where possible) any goals for metrics the software project(s) are expected to reach upon completion of the grant (maximum of 500 words)

Our workplan describes four high-level development objectives, each of which has two to four milestones and deliverables. We list these below, noting the primary developer and their expected completion time (from the start of the funding period).

1.1: New website / documentation [Abernathey, Month 6]
1.2: New interactive examples and tutorials [Abernathey, Month 6]
1.3: User survey [Hamman, Months 1 and 12]

2.1: Explicit indexes refactor [Bovy, Month 9]
2.2: New Index objects for advanced indexing [Bovy, Month 12]
2.3: Index plugin tooling [Bovy, Month 12]

3.1: Abstract Dataset Backend API [Contractor, Month 3]
3.2: Plugin interface for 3rd party backends [Contractor, Month 6]
3.3: Breakout lesser-used backends into 3rd party packages [Contractor, Month 12]

Our primary goal is to make Xarray more useful to scientists in the biomedical sciences. With this in mind, our proposed “user survey” will provide invaluable data that will help us tune our development work and the evaluation metrics while we complete the project. We will also track progress in our milestones using a range of traditional metrics such as user visits to our online

documentation and closure of GitHub issues. Those metrics are listed below for each high-level development objective:

1. Engagement
    a. Number of monthly users as measured by visits to Xarray's online documentation
    b. Number of domain-specific examples in the documentation
    c. Number of GitHub stars and dependents ("Used By")
    d. Citation of Xarray paper in scientific literature
2. Flexible Grids and Indexes
    a. Explicit indexes in Xarray's data-model and development of plugin API (GH1094, GH1603, GH1961)
    b. Development of sample custom indexes such as a KD-Tree -backed index (GH475, GH486) and out-of-core indexes (GH1094,GH1650).
3. Flexible Storage Backends
    a. Refactor storage backends (GH1970, GH2061)
    b. Number of third party backends developed
    c. Documented decrease in backend source code complexity (e.g. maintainability index)

12. **Existing Support (required):**

    List current and recent financial support for the software project(s), including duration, amount in USD, and source of funding (maximum of 250 words)

Xarray developers have received support for various development activities, mostly related to specific scientific applications and the Pangeo Project. Relevant past and present funding includes:

- NSF-OCE 1740633 (PIs Abernathey/Hamman) provided funding for Joe Hamman (approx. 10%) and Matt Rocklin (formerly Anaconda) to improve the integration of Dask within Xarray. The award provided a total of $1.2M in funding between September 2017 and August 2020.
- NASA-ACCESS 80NSSC18M0156 (PI Hamman) provided funding for Joe Hamman (approx. 10%) and Tom Augspurger (Anaconda) to enable Dask and xarray applications for cloud computing. This award comes to an end in mid-2020. This award is also funding Deepak Cherian at approx. 10% to continue general maintenance and development of Xarray. The award provided a total of $1.5M in funding between September 2018 and September 2020.
- NSF-GEO-AGS 1928374 (PI Hamman) is a new award (began Fall 2019) that will provide some resources to Xarray and Jupyter development (approx. 10%). The award will provide a total of $1.5M in funding between September 2019 and August 2022.
- Moore Foundation Climate Data Science Lab (PI Abernathey) - The Moore Foundation made a $2.3M award to Abernathey to develop a Climate Data Science Lab

(11/2019-10/2022). Building on the Pangeo project, this award focuses on developing high-performance domain-specific computational modules for ocean and climate research to enable specific scientific objectives. This code will leverage Xarray extensively but won't necessarily drive core Xarray development.

13. **Landscape Analysis (required):**

    Describe the other software tools (either proprietary or open source) that the audience for this proposal is primarily using. How do the software projects in this proposal compare to these other tools in terms of size of user base, usage, and maturity? (maximum of 250 words)

Today, most of the audience for Xarray uses either lower level data structures (e.g., unlabeled multi-dimensional arrays in Python, R or MATLAB), tabular data structures (e.g., dataframes) or custom domain-specific tooling. Examples of domain-specific tools include Iris, which is quite similar to Xarray but scoped exclusively for climate data, Scikit-Bio, a Python library for working with biological data, and Scipp, a label-aware ND-array package for working with neutron-scattering data. In our view, such packages have failed to obtain critical mass for sustainability precisely because of their narrower scope.

Xarray's advantage is that it solves a very general need in scientific data analysis, without becoming bloated with domain-specific functionality. This is evidenced by recent the recent development of domain specific tools that leverage Xarray:

- Arviz builds on Xarray to provide a toolkit for exploratory data analysis and visualization of Bayesian models,
- Napari, a fast, interactive, multi-dimensional image viewer (developed by CZI) with limited support for visualizing Xarray objects, and
- The Allen SDK which provides tooling for reading and processing Allen Institute for Brain Science data.

A broader set of domain agnostic tools are also beginning to use labeled arrays. For example, Scikit-Learn, a popular machine learning library in Python, is considering adding support for named features (i.e. SLEP18). Additionally, PyTorch, an open source deep learning framework, recently added experimental support for named tensors. We feel that these developments represent a groundswell of interest in labeled multidimensional arrays and that Xarray is well positioned to meet the needs of these applications.

14. **Diversity, Equity, and Inclusion Statement (required):**

    Advancing DEI is a core value for CZI and we are requesting information on your efforts in this area. Describe any efforts the software project(s) named in this proposal have undertaken to increase diversity, equity, and inclusion with respect to their contributors

and audience. Please see examples from successful first cycle applications (maximum of 250 words)

The Xarray has adopted a Code of Conduct derived from the popular Contributor Covenant. In this document, which is included in the Xarray source code repository, we make this pledge:

> "In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation."

The Xarray development team has, like many open source software projects, evolved organically. As such, it suffers from many of the diversity challenges that are well known in the open source software community. We are committed to doing our best to reverse this counterproductive pattern. In particular, we have engaged in the following activities:

- Worked with NumFOCUS to provide recognition to new contributors each year,
- Provided direct mentoring to individuals from underrepresented groups help participation with the project, and
- Coordinated sprints at events like SciPy for beginners.