



**Distributed
visualization of
gridded geophysical
data**

K. A. Endsley and
M. G. Billmire

Distributed visualization of gridded geophysical data: a web API for carbon flux

K. A. Endsley^{1,2} and M. G. Billmire¹

¹Michigan Tech Research Institute (MTRI), Michigan Technological University, Ann Arbor, MI, USA

²School of Natural Resources and Environment (SNRE), University of Michigan, Ann Arbor, MI, USA

Received: 24 June 2015 – Accepted: 30 June 2015 – Published: 23 July 2015

Correspondence to: K. A. Endsley (endsley@umich.edu)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

⏪

⏩

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Abstract

Due to the proliferation of geophysical models, particularly climate models, the increasing resolution of their spatiotemporal estimates of Earth system processes, and the desire to easily share results with collaborators, there is a genuine need for tools to manage, aggregate, visualize, and share datasets. We present a new, web-based software tool – the Carbon Data Explorer – that provides these capabilities for gridded geophysical datasets. While originally developed for visualizing carbon flux, this tool can accommodate any time-varying, spatially explicit scientific dataset, particularly NASA Earth system science Level III products. In addition, the tool’s open-source licensing and web presence facilitate distributed scientific visualization, comparison with other datasets and uncertainty estimates, and data publishing and distribution.

1 Introduction

Today’s scientific enterprise must consider the challenges and opportunities associated with the growing scale of scientific observations, the need for scalable analyses, and the benefits and obligations of sharing scientific outputs. In geophysical models and earth observation science, in particular, a wealth of observations can be generated or collected but rich, collaborative insight requires additional frameworks and software tools. Hence, there is a renewed emphasis in the Earth system sciences on tools and best practices for the documentation and sharing of analyses, metadata generation (e.g., Earth System Documentation, ES-DOC), and scientific provenance (e.g., The Kepler Project; Altintas et al., 2004).

In this paper, we describe a new, web-based framework for managing, analyzing, and collaboratively visualizing Earth system science datasets: the Carbon Data Explorer (<http://spatial.mtri.org/flux-client/>), version 0.2.3. Although the tool’s intended use is for carbon science datasets (e.g., regional carbon flux, global carbon concentration), the Carbon Data Explorer is compatible with any time-varying, spatially explicit Earth

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



system dataset or model output (e.g., land surface temperature, evapotranspiration, aerosol optical thickness). We present the tool as a prototype system that addresses the challenges of increasing scientific data volumes, the need for online analysis, and the desire to share results with collaborators.

Commensurate with the growth of computing power, geophysical models and earth observation systems are producing data with increasingly fine spatial and/or temporal resolution (Nativi et al., 2015). Considering the spatial and temporal dimensions within a dataset simultaneously can be demanding both on computational resources and on a scientist's ability to manage and visualize results. As a conceptual aid, a spatiotemporal dataset consisting of only one parameter of interest can be visualized as a three-dimensional "data cube" (Fig. 1), a representation commonly used in scientific computing (Alder and Hostetler, 2015). The data cube representation has also gained traction in recent scientific visualization tools; UV-CDAT (Santos et al., 2013) and Panoply (Schmunk, 2015) are two examples.

The Carbon Data Explorer also adopts the data cube as a functional interface for high-volume spatiotemporal data. A map view of a single point in time can be visualized as "slicing" the data cube perpendicular to the time ("T") axis and parallel to the geographic ("X-Y") plane. Conversely, a time series display at one point in space (one pair of geographic coordinates) can be visualized as a narrow threading along the time axis and perpendicular to the X-Y plane. For multivariate data we must begin to construct and think in terms of higher-dimensional data "hypercubes". The Carbon Data Explorer is agnostic as to the type of data contained in data cubes and can simultaneously accommodate any number of variables.

While data cubes work well for storing scientific data offline, web browsers and web applications are designed to work largely with plain text documents (interpreted variously as HTML, XML, JavaScript, or other documents). Non-text formats can be downloaded directly from an online directory or through File Transfer Protocol (FTP). Indeed, many scientists, unable to procure or unaware of a more sophisticated solution, provide large collections of outputs directly through FTP – essentially a networked folder

**Distributed
visualization of
gridded geophysical
data**

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures



Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

by best practices in collaborative science. The next phase of the Climate Model Inter-comparison Project, CMIP6, will for the first time involve distributed analyses of climate scenarios (Meehl et al., 2014). Thus, the ability to share and compare model results should motivate the development of web-compatible scientific datasets.

In response to this need, the Carbon Data Explorer allows data providers to share scientific datasets, analyses, and visualizations directly on the web. A data provider might be a modeler, the principal investigator of an interdisciplinary research team, or a technician or information technology (IT) professional embedded in a research team. NASA estimates that these scientists and model developers spend more than 60% of their time preparing model inputs and model inter-comparisons (as cited by Rood and Edwards, 2014). The Carbon Data Explorer was designed specifically to lower or eliminate barriers to bringing scientific results online and making comparisons.

The scientific datasets supported include any gridded or non-gridded time-varying, spatially explicit data that can be decomposed into one variable at a time. The canonical example of a supported dataset is any NASA Level III scientific data product, defined as “variables mapped on uniform space–time grid scales” (NASA, 2010). These geophysical variables are usually derived from Earth observation satellites or models, reanalysis datasets and global or regional Earth system models. Many scientific datasets, particularly Level III products, are already stored as binary (“flat”) files or in complex, hierarchical data structures (e.g. NetCDF or HDF) that were designed to accommodate data cubes (Blower et al., 2013).

In common with the Earth System Grid Federation (Williams et al., 2009), the Carbon Data Explorer aims to provide a common environment for the access to and analysis and visualization of Earth system science datasets. We expect that these and other features of the Carbon Data Explorer make it a useful contribution to the emerging frameworks for data analysis and intercomparison. The remainder of the paper discusses these and other technical details and describes the full suite of features available.

**Distributed
visualization of
gridded geophysical
data**

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

2 Technical description

2.1 Data sources

In the development and evaluation of the tool, we relied heavily on some reference datasets exemplary of those we intend to support. These included a 1°-by-1° carbon flux estimate at 3-hour time steps from the NASA Carnegie Ames Stanford Approach (CASA) model run with Global Fire Emissions Dataset (GFED) input data and 1°-by-1° carbon concentration (XCO_2) data at 6-day time steps modeled by the Carnegie Institution for Science's Department of Global Ecology at Stanford University (<http://dge.stanford.edu/labs/michalaklab/CO2DAAD/>). The CASA-GFED model outputs included monthly uncertainty estimates; the XCO_2 data were gridded by kriging from bias-corrected XCO_2 retrievals.

2.2 Implementation

The Carbon Data Explorer has three main components: a Python application programming interface (API) for data management, a web server API, and a client-side JavaScript web application (Fig. 2). From a data provider's perspective, data enter a pipeline from creation to visualization on the web beginning with the Python API, which transforms and stores the data in a database. The data are then automatically available on the web (or a local area network) through the server API and can be viewed and shared through the web application. This suite of software is designed to run on a single computer or separately on multiple computers, each running any UNIX-like operating system (Mac OS X or a GNU/Linux system).

The Python programming language (version 2.7) was chosen as the framework for data management, manipulation, and storage due to its high-level language design, wide adoption in the scientific community, and available open-source libraries. In particular, as many scientific products are stored as hierarchical data files (HDF) or early Matlab files, Python provides fast and robust support for reading scientific data products

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

⏪

⏩

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



through the NumPy (Van Der Walt et al., 2011) and SciPy (Jones et al., 2015) libraries. We also expect that Python provides an environment that many data providers are already familiar with or can learn easily should they need to extend the data management API to support new or customized datasets.

The web server and web client are both implemented in JavaScript. This was a strategic but also practical decision. JavaScript is fast and expressive. It is also the de facto language of the web; the only language that is natively supported by every modern web browser (Crockford, 2008). While JavaScript is not widely used for scientific computing, no experience with the language is needed to use the Carbon Data Explorer. We selected Node.js (<http://nodejs.org/>) as the framework for running a JavaScript server because it provides event-driven request handling, which, like multithreading, can significantly speed up server response time for most web applications (Tilkov and Vinoski, 2010).

2.3 Data management and storage

Open data APIs for science capitalize on storing and sharing text-based metadata associated with scientific data that are stored in a binary or hierarchical format. We took this a step further and designed a data model that is text-only; that is, the format of the data both on-disk and when transmitted over the web is plain text. Specifically, the data are stored and transmitted as JavaScript Object Notation (JSON) documents. This approach not only ensures compatibility with web browsers but also slightly compresses the data. These JSON documents are stored in a MongoDB database instance which handles indexing and retrieval of such plain-text representations reasonably well. In addition, MongoDB features an aggregation pipeline, which allows us to make sophisticated queries such as “net carbon flux over the last 16 days.” The web server API, which facilitates connections to the MongoDB instance, contains libraries that enable further sophistication with queries, applying fast arithmetic operations for queries such as “the difference between carbon concentration (in ppm) today and this day last year.”

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

⏪

⏩

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Scientific data in the Carbon Data Explorer are conceived of as belonging to a particular run of a “scenario”, i.e., a specific geophysical modeling objective. Each scenario has one timeline associated with it and gridded data belonging to that scenario are uniquely keyed by their date and time. Non-gridded data are assigned arbitrary unique identifiers, making it possible to have two pieces of non-gridded data that represent the same instance in time (or span of time) associated with the same scenario. The gridded data in a scenario must also share the same uniform, rectangular grid. This allows data values to be stored and transmitted independent of the spatial reference information, compressing the data storage and stream to levels that allow for rapid retrieval and display on the web. The “ $X-Y$ ” values associated with gridded data – the spatial coordinates of each data point – are stored separately and transmitted only once to the web application, eliminating redundancy associated with viewing multiple points in the time series. In contrast, non-gridded data are stored with their $X-Y$ values and transmitted as GeoJSON, a spatially explicit form of JSON, as their spatial structure may vary.

Users can shuttle scientific data into and out of the MongoDB instance by directly interacting with the Carbon Data Explorer Python API classes or by using a set of accompanying command line tools designed to ease workflow. Command line tools are available for querying database contents as well as for loading, renaming, and removing datasets from the database. When loading a dataset, its metadata must be specified either via command line argument or via an accompanying JSON file. Examples of required metadata parameters include column identifiers, grid resolution, units, starting timestamp, and time step length. These metadata parameters inform the correct methods for transforming and querying the data for use within the web server API.

The transformation of data from binary or hierarchical flat files to a database representation is facilitated by two Python classes, Models and Mediators, which are loosely based on the Transformation Interface described by Bulka (2001). The Model class is a data model that describes what a scientific dataset looks like; whether it is a time series of gridded maps or a covariance matrix, for instance. The Mediator class describes how a given Model should be read from and the data it contains translated to

a database representation. Some basic Mediator and Model classes are provided in the Python API. It is expected that data providers with a particular output format can easily create new Model and Mediator subclasses to seamlessly read and write data to and from the MongoDB database and the files in which their data are currently stored.

2.4 Provision of scientific data on the web

The Carbon Data Explorer web server API is designed to work out-of-the-box so that data can be served and visualized with the web application on any web browser connected to the same local area network. That is, any user on the same network as the computer running the server can access the Carbon Data Explorer through its internet protocol (IP) address in their web browser. Data providers might choose to host the Carbon Data Explorer locally so as to keep their data private and collaborate internally. Deploying the server and web application on the public web is also easy, though it may require some familiarity with networking technology.

The web server makes data available as resources that are each associated with a uniform resource identifier (URI). The model used for organizing these resources in a single namespace (i.e., under a single host or domain name) is the Representational State Transfer (REST) model (Fielding and Taylor, 2000), in which different representations of data are provisioned with semantics. For example, a list of all available scenarios can be obtained at, e.g., “/scenarios.json” as a JSON document. Alternately, the metadata for a single scenario, e.g., the “casa_gfed_2004” scenario, can be obtained at “/scenarios/casa_gfed_2004.json”. As another example, a map of carbon flux on 18 January 2004 at 03:00 UTC from the “casa_gfed_2004” scenario can be obtained at “/scenarios/casa_gfed_2004/xy.json?time=2004-01-18T03:00:00” where “xy” refers to the X–Y values from our data cube (i.e., a geographic map). This distinguishes map data from a time series, which could be requested in JSON format from the “t.json” resource, e.g., “t.json?start=2003-12-22T03:00&end=2005-01-01T00:00&aggregate=mean&interval=daily”.

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

⏪

⏩

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



These limited examples showcase only a small part of the functionality of the web server's API (Table 1). These relatively human-readable URIs allow experienced users to download data directly if preferred. They are also used behind-the-scenes in the web application to programmatically request data as indicated by a user through its graphical user interface (GUI).

3 Features

In the Carbon Data Explorer client application, a rich user interface (Fig. 3) provides users many options for visualizing, exploring, comparing, and ultimately sharing geophysical data that have been previously imported with the Python API and made available to the client through the web server API. In Table 2 and in the subsequent text, we highlight some of the chief features available to users. A demonstration video (doi:10.5281/zenodo.18941) of the web browser application can also be seen through a link on the project website (<http://spatial.mtri.org/flux/>).

3.1 Spatial visualization and analysis

The default view in the Carbon Data Explorer client application is the “Single Map View” which displays a geographic view (an “ X - Y slice”) of the data at a particular time. The Map Settings define the map projection used (currently a choice between Equirectangular or Mercator) and what kind of basemap should be drawn (e.g., continents with or without political boundaries). When gridded data are drawn on the map, the Symbology options allow a user to specify a color palette from a selection of colorblind-safe, perceptually linear color scales designed by Brewer (2014). Both sequential and diverging color scales are available for linear data that are either constantly increasing or are diverging from a threshold or mean value, respectively. The number of bins in the color scale can also be specified. While the default stretch of the data to the color scale is a standard deviation about the mean, both the measure of central tendency and the

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



model results at a rate of a few hours, days, or months every second (depending on the temporal resolution of the data). A slower animation speed can be selected for a more moderate pace. This high data throughput is made possible by the text-based data format and compression discussed earlier. Aggregates and differenced data can also be animated in time.

A line plot at the bottom of the map shows the “global” time series for the currently viewed scenario by default; it is the aggregate mean value across the X – Y domain at each point in time. This provides an overview of the overall trend in the data across the spatial domain. When a spatial filter is applied, an aggregate time series for only that region can be generated. The non-aggregate time series for a specific pixel can also be obtained by clicking on that grid point in the map. Retrieval of a time series data for the line plot is slower than other data requests but it still returns results in seconds.

3.3 Multiple-time and multiple-model comparison

The “Coordinated View” allows comparison of multiple adjacent map views; it is essentially a grid of multiple Single Map View elements. These maps synchronize their extent whenever the user pans or zooms so that the same portion of the globe is displayed in each one. The user’s cursor will now display not just the value of a data point in one map but the value at that those spatial coordinates in every map facilitating pixel-to-pixel comparison across the maps. Up to nine (9) maps can be viewed at once which allows for nine different time points or nine different models to be viewed simultaneously.

3.4 Other features

A user’s Map Settings, Symbology, and other global settings are stored in the web browser so that, upon closing the browser and returning to the web application later, the same color scale, map projection, and other settings are automatically applied. This allows users to customize their view of a dataset and their workspace within the tool.

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

⏪

⏩

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



All of these settings can also be encoded as a URI (or URL). This allows specific views of a dataset to be “bookmarked” or shared with others over the web. With this feature, a user can apply a specific color scale, stretch (or threshold to highlight a particular anomaly) or an aggregate or differenced model result and then share a link that ensures that their team member will see the data exactly the same way. For offline storage and sharing of results, model visualizations and data slices can be exported as image files, CSVs (for non-gridded data), or as geospatial data (for gridded data) in the form of ESRI ASCII Grid files or GeoTIFFs; the latter two formats enable model results to be downloaded and opened in a desktop GIS like ArcGIS or QGIS.

4 Concluding remarks

The Carbon Data Explorer is presented as a prototype for a comprehensive data management, analysis, visualization, and sharing framework for Earth system science datasets, particularly gridded spatiotemporal datasets (e.g., NASA Level III data products). With its unique text-based data representations, gridded scientific datasets can be rapidly manipulated, analyzed, and displayed on the web. In response to the new protocols of CMIP6, the Carbon Data Explorer provides a framework for the distributed analysis of climate model outputs. The framework’s open source licensing and web integration enable the visualization and sharing of scientific data through either a secure network or public portal. It is hoped they will also facilitate the future improvement of the Carbon Data Explorer and the inspiration of similar and better tools for Earth system science.

Code availability

The source code is available from GitHub (<https://github.com/MichiganTechResearchInstitute/CarbonDataExplorer>) under the MIT license. A built version of the web browser application (“flux-client”) is available upon request. This

can significantly help integration and deployment of the visualization and analysis front-end.

**The Supplement related to this article is available online at
doi:10.5194/gmdd-8-5741-2015-supplement.**

5 *Acknowledgements.* The software described in this paper was developed by the Michigan Tech Research Institute (MTRI) in partnership with Anna Michalak at the Carnegie Institution for Science's Department of Global Ecology at Stanford University and with funding from NASA (Grant #NNX12AB90G). The authors would also like to acknowledge the contributions of Nicholas Molen to the early development of the Carbon Data Explorer and of Reid Sawtell to the front-end web browser application. Special thanks go to Mae Qiu at Stanford University and Vineet Yadav at NASA Jet Propulsion Laboratory for their contributions to design and testing. The authors would also like to thank James Arnott and Scott Kalafatis at the School of Natural Resources and Environment at the University of Michigan for their reviews of an early draft of this paper.

15 **References**

- Alder, J. R. and Hostetler, S. W.: Web based visualization of large climate data sets, *Environ. Modell. Softw.*, 68, 175–180, doi:10.1016/j.envsoft.2015.02.016, 2015. 5743, 5744
- Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., and Mock, S.: Kepler: an extensible system for design and execution of scientific workflows, in: *Proceedings 16th International Conference on Scientific and Statistical Database Management*, Santorini Island, Greece, 21–23 June 2004, 423–424, doi:10.1109/SSDM.2004.1311241, 2004. 5742
- 20 Blower, J. D., Gemmill, A. L., Griffiths, G. H., Haines, K., Santokhee, A., and Yang, X.: A web map service implementation for the visualization of multidimensional gridded environmental data, *Environ. Modell. Softw.*, 47, 218–224, doi:10.1016/j.envsoft.2013.04.002, 2013. 5744, 5745
- 25 Brewer, C. A.: Color Brewer 2.0, available at: <http://www.colorbrewer.org> (last access: 24 June 2015), 2014. 5750

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures



Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Bulka, A.: Transformation Interface Design Pattern, available at: <http://www.andypatterns.com/files/74091232001410AndyBulkaTransformationInterfacePattern.pdf> (last access: 24 June 2015), 2001. 5748

Cornillon, P., Gallagher, J., and Sgouros, T.: OPeNDAP: Accessing data in a distributed, heterogeneous environment, *Data Science Journal*, 2, 164–174, doi:10.2481/dsj.2.164, 2003. 5744

Crockford, D.: JavaScript: The Good Parts, 1st Edn., O'Reilly Media, Inc., doi:10.1241/johokanri.44.584, 2008. 5747

Fielding, R. T. and Taylor, R. N.: Principled design of the modern Web architecture, in: ICSE '00: Proceedings of the 22nd International Conference on Software Engineering, 407–416, doi:10.1145/337180.337228, 2000. 5749

Jones, E., Oliphant, T. E., and Peterson, P.: SciPy: Open source scientific tools for Python, available at: <http://www.scipy.org/> (last access: 24 June 2015), 2015. 5747

Meehl, G. A., Moss, R., Taylor, K. E., Eyring, V., Stouffer, R. J., Bony, S., and Stevens, B.: Climate model intercomparisons: preparing for the next phase, *EOS T. Am. Geophys. Un.*, 95, 77–78, doi:10.1002/2014EO090001, 2014. 5745

NASA: Data Processing Levels for EOSDIS Data Products, available at: <http://science.nasa.gov/earth-science/earth-science-data/data-processing-levels-for-eosdis-data-products/> (last access: 24 May 2015), 2010. 5745

Nativi, S., Mazzetti, P., Santoro, M., Papeschi, F., Craglia, M., and Ochiai, O.: Big Data challenges in building the Global Earth Observation System of Systems, *Environ. Modell. Softw.*, 68, 1–26, doi:10.1016/j.envsoft.2015.01.017, 2015. 5743

Rood, R. B. and Edwards, P. N.: Climate Informatics: Human Experts and the End-to-End System, *Earthzine*, May, available at: <http://earthzine.org/2014/05/22/climate-informatics-human-experts-and-the-end-to-end-system/> (last access: 24 June 2015), 2014. 5745

Santos, E., Poco, J., Wei, Y., Liu, S., Cook, B., Williams, D. N., and Silva, C. T.: UV-CDAT: analyzing climate datasets from a user's perspective, *Comput. Sci. Eng.*, 15, 94–103, doi:10.1109/MCSE.2013.15, 2013. 5743

Schmunk, R. B.: Panoply netCDF, HDF and GRIB Data Viewer, available at: <http://www.giss.nasa.gov/tools/panoply/>, last access: 24 June 2015. 5743

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

Title Page

Abstract

Introduction

Conclusions

References

Tables

Figures

◀

▶

◀

▶

Back

Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion



Sun, X., Shen, S., Leptoukh, G. G., Wang, P., Di, L., and Lu, M.: Development of a web-based visualization platform for climate research using Google Earth, *Comput. Geosci.*, 47, 160–168, doi:10.1016/j.cageo.2011.09.010, 2012. 5744

Tilkov, S. and Vinoski, S.: Node.js: using JavaScript to build high-performance network programs, *IEEE Internet Comput.*, 14, 80–83, doi:10.1109/MIC.2010.145, 2010. 5747

Van Der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy array: a structure for efficient numerical computation, *Comput. Sci. Eng.*, 13, 22–30, doi:10.1109/MCSE.2011.37, 2011. 5747

Williams, D. N., Ananthakrishnan, R., Bernholdt, D. E., Bharathi, S., Brown, D., Chen, M., Chervenak, A. L., Cinquini, L., Drach, R., Foster, I. T., Fox, P., Fraser, D., Garcia, J., Hankin, S., Jones, P., Middleton, D. E., Schwidder, J., Schweitzer, R., Schuler, R., Shoshani, A., Siebenlist, F., Sim, A., Strand, W. G., Su, M., and Wilhelmi, N.: The earth system grid: enabling access to multimodel climate simulation data, *B. Am. Meteorol. Soc.*, 90, 195–205, doi:10.1175/2008BAMS2459.1, 2009. 5745

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

[Title Page](#)

[Abstract](#)

[Introduction](#)

[Conclusions](#)

[References](#)

[Tables](#)

[Figures](#)

[⏪](#)

[⏩](#)

[◀](#)

[▶](#)

[Back](#)

[Close](#)

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

Table 1. Entry points for the Carbon Data Explorer web server API.

| Web Server API Entry Point | Description |
|------------------------------------|---|
| <code>/scenarios.json</code> | Requests metadata for all or selected datasets |
| <code>/[scenario]/grid.json</code> | Requests a GeoJSON representation of the scenario's X–Y grid |
| <code>/[scenario]/xy.json</code> | Requests data values corresponding to the scenario's spatial grid |
| <code>/[scenario]/t.json</code> | Requests time series data |

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

[Title Page](#)

[Abstract](#) [Introduction](#)

[Conclusions](#) [References](#)

[Tables](#) [Figures](#)

[⏪](#) [⏩](#)

[◀](#) [▶](#)

[Back](#) [Close](#)

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

Table 2. List of features (present when marked with an “X”) in the two visualization modes of the Carbon Data Explorer web browser application.

| Category | Feature | Single Map View | Coordinated View |
|-----------|---|-----------------|------------------|
| Mapping | Gridded data map display | x | x |
| | Non-gridded data map display | x | |
| | Basemap selection | x | x |
| | Map projection selection | x | x |
| | Map zoom and pan | x | x |
| | View multiple map frames simultaneously | | x |
| | View global/continental/regional data | x | x |
| Symbology | Color palette selection | x | x |
| | Scaling, stretching, and thresholding | x | x |
| Analysis | Animation | x | |
| | Map pixel querying | x | x |
| | Temporal queries and aggregation | x | |
| | Spatial queries and aggregation | x | |
| | Time series line plot | x | |
| | Display difference of two maps | x | |
| | Side-by-side comparison | | x |
| Sharing | Data export (as image or GIS file) | x | |
| | Browser remembers settings | x | |
| | Shareable URI/URL generation | x | |



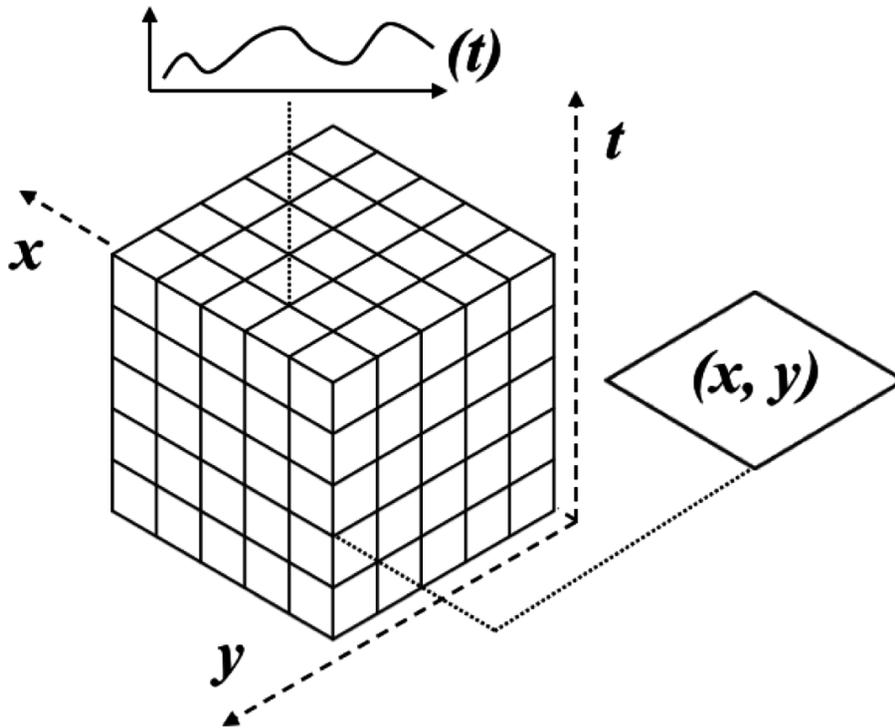


Figure 1. A three-dimensional “data cube” in which spatial data of two dimensions (e.g., latitude and longitude) are combined with a third dimension of time. In this view, a horizontal slice perpendicular to the time (t) axis corresponds to a geographic map while a line parallel to the time (t) axis represents a time series.

**Distributed
visualization of
gridded geophysical
data**

K. A. Endsley and
M. G. Billmire

| | |
|--------------------------|--------------|
| Title Page | |
| Abstract | Introduction |
| Conclusions | References |
| Tables | Figures |
| ◀ | ▶ |
| ◀ | ▶ |
| Back | Close |
| Full Screen / Esc | |
| Printer-friendly Version | |
| Interactive Discussion | |



Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

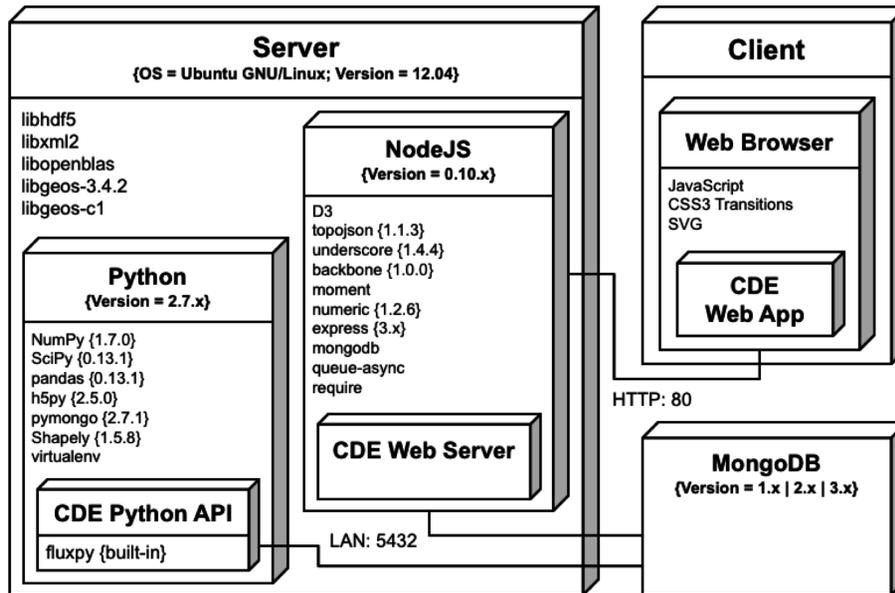


Figure 2. A unified modeling language (UML) deployment diagram for the Carbon Data Explorer (CDE), illustrating the configuration and connections between the components as currently deployed.

| | |
|--------------------------|--------------|
| Title Page | |
| Abstract | Introduction |
| Conclusions | References |
| Tables | Figures |
| ◀ | ▶ |
| ◀ | ▶ |
| Back | Close |
| Full Screen / Esc | |
| Printer-friendly Version | |
| Interactive Discussion | |

GMDD

8, 5741–5761, 2015

Distributed visualization of gridded geophysical data

K. A. Endsley and
M. G. Billmire

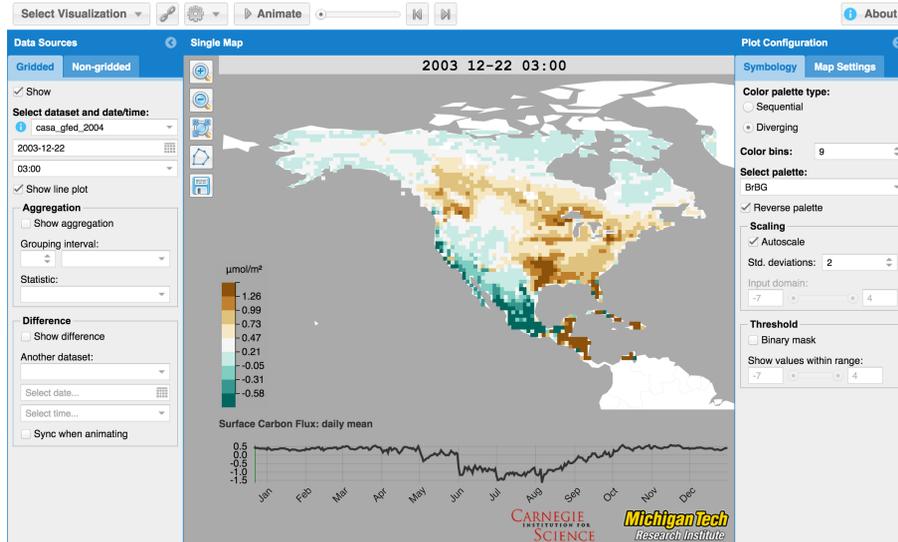


Figure 3. Screenshot of the Carbon Data Explorer web browser application in the “Single Map View” mode.

[Title Page](#)

[Abstract](#) [Introduction](#)

[Conclusions](#) [References](#)

[Tables](#) [Figures](#)

[◀](#) [▶](#)

[◀](#) [▶](#)

[Back](#) [Close](#)

[Full Screen / Esc](#)

[Printer-friendly Version](#)

[Interactive Discussion](#)

