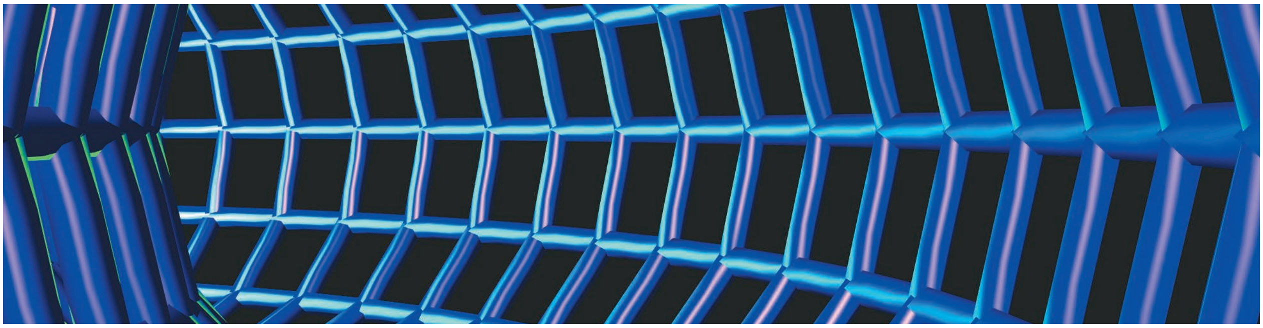


Ultrascale Visualization of Climate Data

Ultrascale Visualization Climate Data Analysis Tools Project Team

Collaboration across research, government, academic, and private sectors is integrating more than 70 scientific computing libraries and applications through a tailorable provenance framework, empowering scientists to exchange and examine data in novel ways.

Fueled by exponential increases in the computational and storage capabilities of high-performance computing platforms, climate simulations are evolving toward higher numerical fidelity, complexity, volume, and dimensionality. These technological breakthroughs are coming at a time of exponential growth in climate data, with estimates of hundreds of exabytes by 2020.[1]

To meet the challenges and exploit the opportunities that such explosive growth affords, a consortium of four national laboratories, two universities, a government agency, and two private companies formed to explore the next wave in climate science. Working in close collaboration with domain experts, the Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) project aims to provide high-level solutions to a variety of climate data analysis and visualization problems:

- *Dealing with big data analytics.* Climate science is no different from other domains in its pursuit of solutions to process, analyze, and visualize massive datasets.
- *Sensitivity analysis.* The community must be able to push ensemble analysis, uncertainty quantification, and metrics computation to new boundaries.
- *Heterogeneous data sources.* Climate science data comes from simulations, observations, and reanalysis. Any visualization and analysis solution must unify these sources.
- *Reproducibility.* All science must support systematic data maintenance by providing provenance to ensure reliable and persistent links between workflows.
- *Multiple disciplinary domains.* Complexity stems from the need to incorporate a broad nexus of climate and other related science domains such as climate adaptation and mitigation for water, energy, and agriculture conservation.
- *Flexible, scalable architecture.* Any unifying structure must be able to incorporate both existing and future software components with minimal or no infrastructure modification.

As the "UV-CDAT Consortium" sidebar describes, the project integrates computational and domain scientists from all sectors in tackling these problems, with the ultimate goal of building an ultrascale data analysis and visualization system that will empower novel data exchanges. The project, which began in late 2010 and is expected to complete in late 2013, has met several major objectives. As of August 2013, the team has

- released the official UV-CDAT system version 1.4 (http://uv-cdat.org);
- addressed projected scientific needs for data analysis and visualization;

Published by the IEEE Computer Society

- extended UV-CDAT to support the latest regridding by interfacing to the Earth System Modeling Framework (ESMF; www.earthsystemmodeling.org) and Climate and Forecast library (LibCF; www.unidata.ucar.edu/software/libcf); and
- successfully supported ongoing climate model evaluation activities for US Department of Energy (DOE)'s climate applications and projects (www.climatemodeling.science.energy.gov/projects), such as the Intergovernmental Panel on Climate Change (IPCC) assessment report and Climate Science for a Sustainable Energy Future (CSSEF).

UV-CDAT brings to bear many capabilities that directly address climate scientists' needs, including parallel streaming statistics, optimized parallel I/O, remote interactive execution, workflow capabilities, and the automatic capture of data provenance. Already, the UV-CDAT architecture is enabling new capabilities in visualization, regridding, and statistical analysis.

UV-CDAT Consortium

UV-CDAT's integrated, cross-institutional effort is unique in its breadth and depth of expertise. Four US Department of Energy (DOE) national laboratories—Lawrence Berkeley (LBNL), Lawrence Livermore (LLNL), Los Alamos (LANL), and Oak Ridge (ORNL)—are focusing on the development of large-scale parallel analytics and the diagnosis of climate model simulation (atmosphere, land, ocean, and sea ice) and observational postprocessing.

Two universities—Polytechnic Institute of New York University and the University of Utah—are responsible for provenance and workflow and streaming visualization.

From the private sector, Kitware is working on software process, cross-platform build and test suites, and development of a spatiotemporal parallel pipeline. Tech-X is supplying regridding and reprojection libraries.

NASA's Goddard Space Flight Center provides 3D data visualization and UV-CDAT tutorials.

ARCHITECTURAL OVERVIEW

One of our main project objectives is to assist in ultrascale scientific analysis and visualization. In light of this goal, one of UV-CDAT's strongest features is its Python-based framework, which integrates disparate technologies under one overarching infrastructure. Figure 1 shows the framework's conceptual design.

Using standard common protocols and application programming interfaces (APIs), the framework allows UV-CDAT to integrate 70-plus (to date) software components. The primary software stack comprises climate data analysis tools (CDAT), VisTrails,[2] Data Visualization 3D (DV3D), and ParaView.[3] DV3D and ParaView are two of five analysis and visualization tools that have gained considerable popularity. The other three are VisIt (https://wci.llnl.gov/codes/visit/home.html), Visualization Streams for Ultimate Scalability (ViSUS; http://visus.us), and Visualization Toolkit (VTK; www.vtk.org).

However, none of these tools are customized for climate research, forcing climate scientists to build individual analyses around each tool, and in the process often recreating code developed by other scientists. CDAT provides a more direct approach because it integrates these tools into the UV-CDAT framework, which essentially customizes them to process climate data and common analyses.

Scientists must also be able to create integrated analysis and visualization pipelines that interweave multiple tools, libraries, and services. To aid that creation, UV-CDAT relies

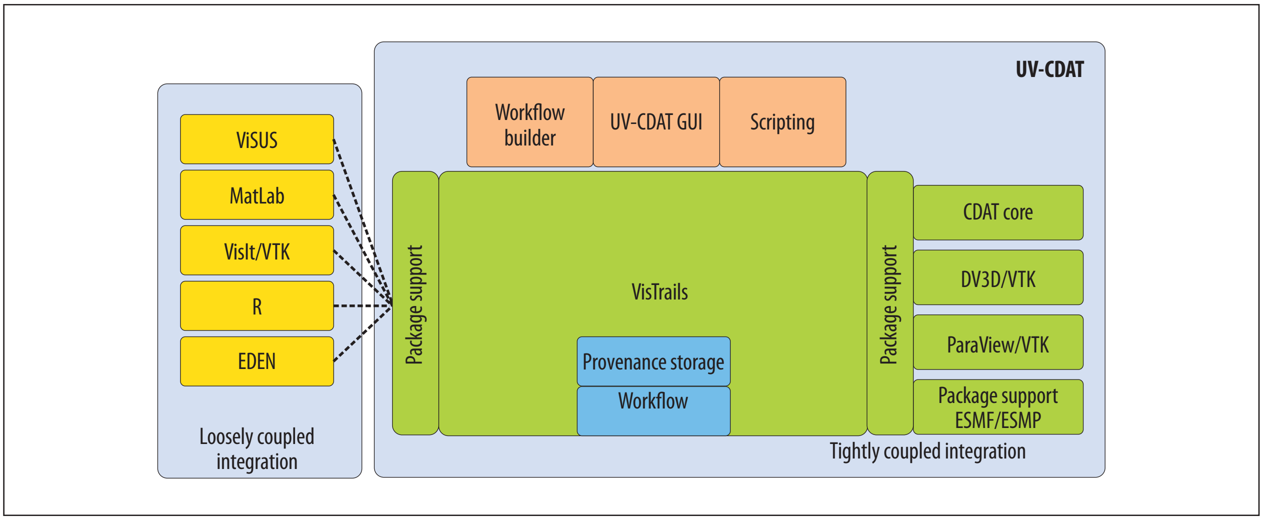


Figure 1. The Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) project framework. By uniting common protocols and APIs in either a loose coupling or tight integration, the framework enables scientists to understand climate and observation data across models, providing the breadth of understanding needed to interpret climate change.

on VisTrails, an open source provenance and workflow system.

Using an open source, object-oriented Python scripting language that is easy to learn, CDAT links disparate software subsystems and packages to form an integrated environment for data analysis. The integration concept is simple and flexible, allowing researchers to interchange the application's parts and expand for the future. Through Python, CDAT provides a full-featured scripting language with a variety of user interfaces, including command-line interactions, standalone scripts (applications), and a GUI.

The CDAT core consists of the Climate Data Management System (CDMS),[4] large-array numerical operations, climate analysis and diagnosis, and statistics. These subsystems provide climate scientists with on-demand analysis and diagnostics, which facilitate a better understanding of climate processes and effects.

> **For climate modeling simulation and, more recently, observation, groups are increasingly storing their data in netCDF, whereas the climate and forecast (CF) methodology is becoming the de facto convention for defining metadata.**

By combining these CDAT core subsystems with enhanced data visualization engines, a provenance workflow system, and an integrated framework, UV-CDAT can offer end-to-end solutions for data management, analysis, and visualization for the ultrascale datasets.

The UV-CDAT framework couples software infrastructures either tightly for greater system functionality and communication with other components or loosely for fast integration. Tightly coupling CDAT's core subsystems with the VTK/ParaView infrastructure, for example, enables seamless high-performance parallel streaming data analysis and visualization of massive climate datasets. Loosely coupled integration then allows the addition of other tools such as Matlab and customized data analysis tools with little or no modification. Within both paradigms, UV-CDAT provides data provenance, workflow capture, and mechanisms to support data analysis via the VisTrails infrastructure.

In addition, users can add custom functionality through an intuitive interface that includes tools for workflow analysis and visualization construction. To augment these capabilities, we include links to the R statistical analysis environment (www.r-project.org) as well as to enhanced visualization tools such as DV3D, Exploratory Data Analysis Environment (EDEN),[5] and VisIt. All these functions and tools are integrated under a Python- or Qt-based architecture (http://qt-project.org).

DATA, METADATA, AND GRIDS

Data formats play an integral role in consolidating geoscience information. In climate research, data comes from model simulations, instruments, or observations as well as metadata, such as how the data was generated, what it represents, and what is to be done with it. Most collections of model runs, observations, and analysis files provide a uniform data access interface to conventional formats, such as the Network Common Data Form (netCDF), Hierarchical Data Format (HDF), Gridded Information in Binary form (GRIB), and Post-Processing (PP) format. For climate modeling simulation and, more recently observation, groups are increasingly storing their data in netCDF, whereas the climate and forecast (CF) methodology is becoming the de facto convention for defining metadata. Combining netCDF and CF into one convention, netCDF-CF, makes it easy for scientists to compare and display a range of geoscience datasets and to decide which quantities are comparable. This capability supports building applications with powerful extraction, regridding, and display functionalities. With UV-CDAT's Climate Model Output Rewriter (CMOR), scientists can easily obtain properly formatted data, which will encourage the smooth adoption of the netCDF-CF convention.

ANALYSIS

Analysis consists of regridding (interpolating data from one grid to another), exploring the data and generating hypotheses, parallel processing, and capturing provenance information.

Regridding

Our recent user survey revealed that regridding is among the most widely used features in climate data analysis tools. In UV-CDAT, we extended this feature to support curvilinear grids, which ocean and atmospheric models often rely on to overcome numerical stability issues at the North and South Poles. Examples of curvilinear grids are the displaced or rotated pole grid and the tripolar grid, which some ocean models use to remove the North Pole singularity from the grid. The block-structured, cubed-sphere grid that some atmospheric models rely on is another example of a curvilinear grid with no singularity at the poles.

Regridding Earth data is a conservative interpolation that presents unique challenges. The data might have missing or invalid values, such as ocean data values that fall on land. Also, although conservative interpolation is the method of choice for cell-centered data, it can be significantly more numerically intensive than nodal interpolation.

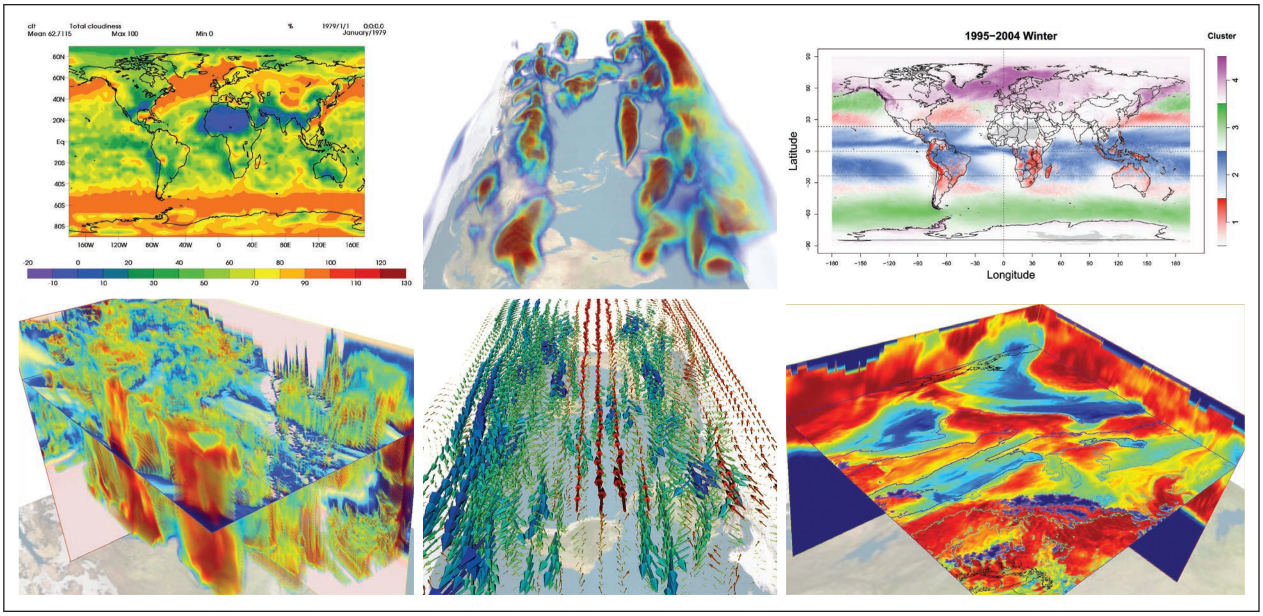


Figure 2. Accessing multiple visualization libraries through the UV-CDAT framework. The framework supports a wide range of 2D and 3D visualization techniques. The images in this collage are products of the CDAT (climate data analysis tools) and DV3D (Data Visualization 3D) libraries.

We have addressed these challenges by leveraging multiple existing interpolation libraries, and by designing a single Python regridding interface that supports multiple interpolation tools, such as ESMF and the Spherical Coordinate Remapping and Interpolation Package (SCRIP; http://climate.lanl.gov/Software/SCRIP), and methods, which at present include linear nodal, quadratic nodal, and conservative. On the basis of grid type (rectilinear or curvilinear) and data type (nodal or cell), the interface automatically selects the most appropriate tool and method for the task.

Exploratory data analysis and hypothesis generation

Another important aspect of the UV-CDAT framework is its ability to provide users with the means to quickly explore massive amounts of data to form new hypotheses and verify simulation data. Through one interface, scientists can now access CDAT, ParaView, VisIt, and DV3D—four important toolkits for visual data exploration, as Figure 2 shows. UV-CDAT thus provides all traditional visualization methods, such as slicing, volume rendering, and isosurfacing, as well as the ability to explore long time series and create animation sequences. Through a spreadsheet paradigm, scientists can combine a variety of 2D and 3D plots.

Parallel processing

As resolution increases, climate models are continuously improving numerical fidelity—resolution and number of variables within a smaller region—to the point that the memory footprint becomes too large for the data to reside on a single processor. Most platforms can accommodate the loading and processing of only one 3D variable at a 10-km resolution in a single time step. To help users handle such memory-heavy processing and other numerically intensive operations, we extended the CDMS arrays[4] in CDAT to allow remote memory access within the scripting UV-CDAT environment. The new functionality supports remote data access through a get method, which takes the remote processing rank and a tuple that uniquely represents a slice of the data to be fetched.

We used the mpi4py module (http://mpi4py.scipy.org) to implement these functions in Python, relying on recent one-sided communication enhancements to the MPI-2 standard. The resulting implementation of distributed array functionality works in any number of dimensions and for multiple data types. Although simple, an implementation based on remote memory access is more flexible than one based on point-to-point send or receive calls because the process that exports data does not need to know which process to send data to, and each process can access data residing on any other processor.

Capturing provenance information

During analysis, VisTrails automatically captures provenance information, making it possible to reproduce and share results, and thereby reducing the effort to manage scripts and data files. Each analysis has a corresponding workflow that is updated when the analysis changes, such as when the user changes a parameter or introduces an intermediate step. Because updates are incremental, VisTrails keeps all versions of the analysis as provenance information. Figure 3a is an example of an automatically

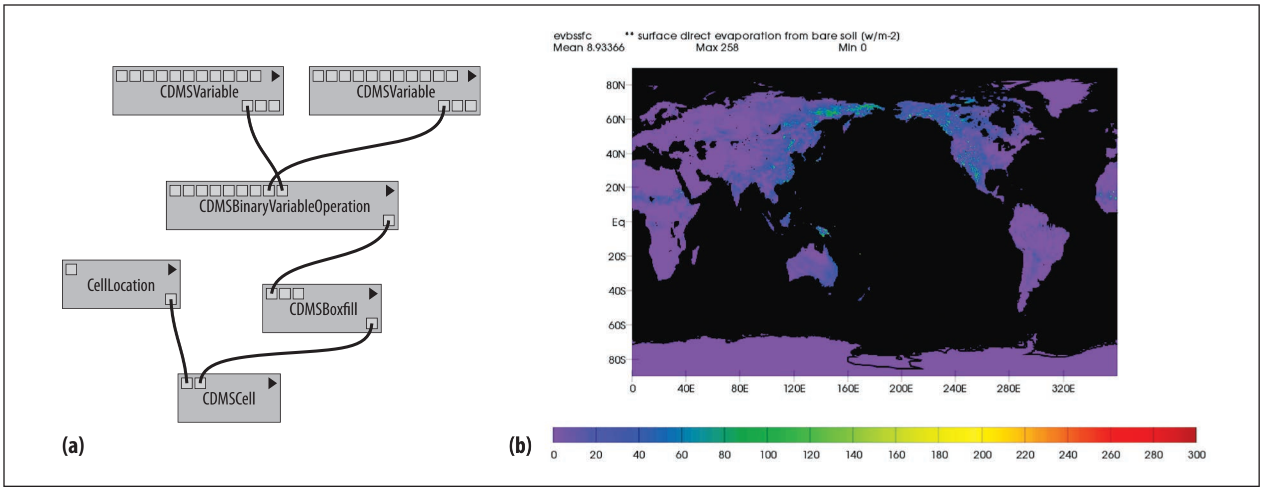


Figure 3. Example showing (a) the UV-CDAT workflow and (b) resulting plot for regridding a variable and plotting it using the boxfill plot type from the CDAT library. The user is applying CDMS (Climate Data Management System) to read in the data, mask out the ocean, and plot the image with metadata information—all functions in CDAT's core subsystems.

generated workflow for regridding a variable and plotting it using the boxfill plot type from the CDAT library. Figure 3b shows the resulting plot.

WORKFLOW GENERATION

In the UV-CDAT framework, workflow is captured and stored through community tools, such as software compilation controllers, dashboards, and testing tools, and analyzed through open source systems that support data exploration and visualization.

Community tools and environments

Developing large-scale software requires a rigorous process that takes a concept through design, development, implementation, code integration, testing, and user delivery in a robust and systematic way. This process requires monitoring to ensure quality and involves tools such as the open source CMake, CTest, and CDash packages.[6] In the UV-CDAT framework, community tools include these packages as well as Github, a repository for data, documentation, and code.

- CMake uses simple platform- and compiler-independent configuration files to control software compilation. A cross-platform build system based on CMake generates native makefiles and workspaces.
- CTest, a testing tool distributed as a part of CMake, automates code update, configuration, build, and test operations.
- CDash, an open source, Web-based software-testing server, aggregates, analyzes, and displays the results of software testing processes that clients submit.

UV-CDAT's software process supports agile development methods and is motivated by test-driven development approaches. A similar process in use by thousands of software systems has scaled to tens of millions of lines of code.[6]

Community visualization and analysis

Visualization and analysis requires systems that support data exploration and visualization, interactive visualization, rapid access to large datasets, postprocessing, and statistical computing, as well as a flexible, encompassing interface.

VisTrails. VisTrails lets users specify the computational processes that will integrate existing applications, loosely coupled resources, and libraries simply by clicking buttons or dragging variables and plot types. VisTrails' API converts these events into workflow operations, such as module creation and parameter changes, and captures them as provenance information. It then notifies the system to update the plots and the GUI, as necessary. Scientists can also use the workflow builder to edit workflows directly and to create new plots.

VisTrails captures and maintains a detailed history of the steps followed and data derived during an exploratory task, maintaining the provenance of the workflows that derive these data products as well as their executions. Using a thin Python interface encapsulated by a set of VisTrails modules, developers can expose their libraries (in any language) to UV-CDAT, making it simple to integrate tools and libraries as well as to quickly prototype new functions.

DV3D. DV3D, a VisTrails package of high-level modules for UV-CDAT, provides user-friendly workflow interfaces for advanced climate data visualization and analysis. DV3D provides the interfaces, tools, and application integration required to make VTK's analysis and visualization power readily accessible to scientists without exposing details such as actors, cameras, and renderers. It can run as a desktop application or distributed over a set of nodes for hyperwall or distributed visualization.

DV3D offers scientists a set of coordinated interactive 3D plot types that provide insightful dataset views. Each DV3D plot type offers a unique perspective by highlighting particular data features, and multiple plots can be combined synergistically to gain a deeper understanding of the natural processes underlying the data.

The *volume slice* plot provides a set of slice planes that can be interactively dragged over datasets so that scientists can quickly and easily browse the 3D structure of datasets, compare variables in 3D, and probe data values. The *volume render* plot maps variable values within a data volume, varying with opacity and color, so that scientists can create an overview of the data topology and see complex 3D structures at a glance. The *Hovmoller volume slice and render* plots operate on a data volume structured with time (instead of height or pressure level). This plot allows scientists to quickly and easily browse the 3D structure of spatial time series.

Additional types include a textured isosurface plot and various vector field plots. Seamless integration with CDAT's CDMS and other analysis tools provides a rich suite of exploratory visualization and analysis methods for addressing climate dataset complexity.

ParaView. ParaView is an open source, multiplatform framework that enables interactive data visualization either locally or remotely, running in standalone or client-server mode. In the standalone mode, ParaView performs data processing and rendering locally on the client; in client-server mode, it performs most of these operations on the server, sending only the geometry or rendered images to the client.

ParaView attacks the problems of visualizing big data through approaches such as parallel processing, client-server separation, and the separation of render and data servers—all are powerful solutions that the UV-CDAT framework exploits by tightly integrating ParaView, allowing the user to create multiple visualizations of the same variable. For example, a user can create a contour and a slice representation of a single variable in the same shared view. Users create a ParaView pipeline by using UV-CDAT's GUI to create a workflow. Once connected, a user can browse the remote file system to select datasets for visualization.

A ParaView-based UV-CDAT spatiotemporal pipeline was developed to meet the critical need for fast scalable timestep processing of high-resolution spatial and temporal climate datasets, thus enabling image sequence production and time averaging of these data.

ViSUS. Dealing with large datasets can rapidly become cumbersome for resources set up to handle smaller-scale data. To deal with this mismatch, UV-CDAT is integrating a new complementary ViSUS-based technology. At its core, ViSUS focuses on providing fast access to extremely large datasets, regardless of cache memory. Exploiting the notion of hierarchical space-filling curves, ViSUS provides a progressive, multiresolution data stream that drastically reduces the file I/O needed to extract information, such as a slice from a 3D dataset.

The ViSUS architecture consists of a visualization client running under various GUI front ends (including a Web browser) and a lightweight server that enclose remote data access; the client is integrated into UV-CDAT.

ViSUS has already demonstrated interactive access to terabytes of simulation data on devices as small as a smartphone, while remotely using low-bandwidth connections such as public wireless high-speed Internet and network (Wi-Fi) hotspots. Such rapid access will enable scientists to easily explore remote datasets directly from a personal desktop before committing to extensive data transfers or remote analysis.

As part of our work on the UV-CDAT framework, we added two climate-specific operations to VisIt: peaks-over-threshold computing and extreme value analysis.

VisIt. VisIt is an open source, turnkey application for visualizing and analyzing large-scale simulation and experimental datasets. Beyond a mechanism for making pretty pictures, VisIt is an infrastructure for parallelized, general postprocessing of massive datasets. Target use cases include data exploration, comparative analysis, visual debugging, quantitative analysis, and presentation graphics.

The basic design is a client-server model with a parallelized server. The client-server architecture permits visualization in a remote setting, while server parallelization allows the interactive processing of large datasets. VisIt users have been able to visualize datasets as large as a structured grid with 216 billion data points and a particle simulation dataset with one billion points, as well as curvilinear, unstructured, and adaptive mesh refinement meshes with hundreds of millions to billions of elements.

Within UV-CDAT, VisIt has a loosely coupled infrastructure—client components are wrapped and integrated within UV-CDAT while the server component executes separately. With this arrangement, VisIt can execute climate analysis algorithms on machines that leverage distributed processing either locally or remotely.

As part of our work on the UV-CDAT framework, we added two climate-specific operations to VisIt: peaks-over-threshold computing and extreme value analysis. Both these operations use GNU-R scripts at their core and interface with VisIt and UV-CDAT through the VTK-R bridge.

Figure 4 shows a computation of the extreme value analysis operation using VisIt-R and temperature data rendering using VisIt, as well as DV3D, CDAT, and ParaView plots. Scientists use the operation and model output and observations to estimate past and future changes in extreme precipitation and other climate variables.

Statistical computing. R is a widely used statistical computing package, so incorporating it into VisIt and UV-CDAT provides access to the many statistical analysis algorithms that are at the core of climate analysis work. VisIt uses custom R scripts to compute several climate-related operations, and we are actively working on expanding R procedures in UV-CDAT.

EDEN. Developed collaboratively with climate researchers on the CSSEF project, EDEN blends interactive information visualization techniques with automated statistical analytics to effectively guide the scientist to the most significant relationships and thus provide visual data mining. EDEN is built on a set of coordinated views with central parallel coordinate visualization, which blends interactive information visualization techniques with automated statistical analytics to effectively guide the scientist to the most significant relationships.

Graphical user interface

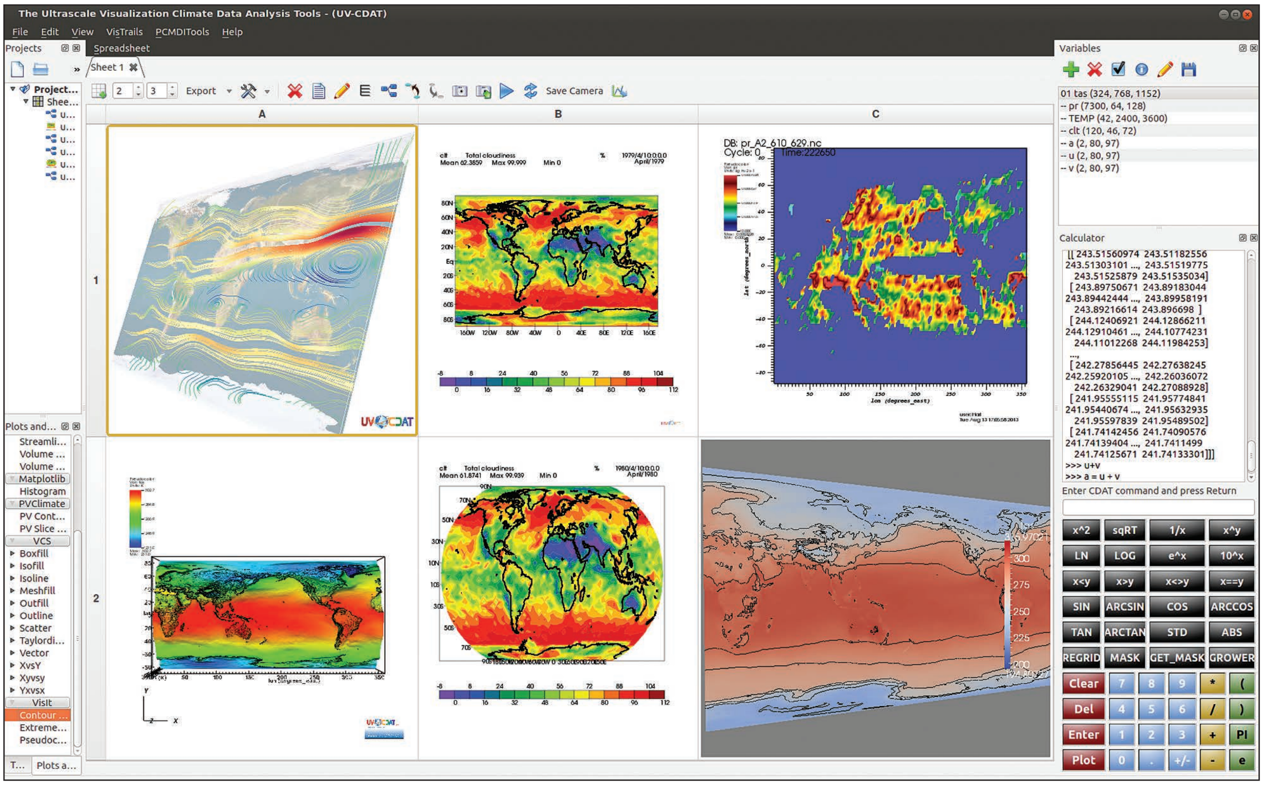
The GUI for the UV-CDAT framework is based on the notion of a VisTrails visualization spreadsheet (the six images in the center of Figure 4) or a resizable grid in which each cell contains a visualization. Spreadsheets maintain their provenance, and can be saved and reloaded. The visualizations are both fully customizable and reproducible for data exploration and decision making.

Around the spreadsheet are the tools for building visualizations. To create a visualization, users drag a variable from the variable panel and a plot type from the plot list to a spreadsheet cell. The project panel (top left) lets users group spreadsheets into projects and name visualizations and spreadsheets. The plot list (bottom left) shows the available plot types, and the variable panel (top right) maintains the loaded data variables. A calculator widget (bottom right) aids users in computations to derive new variables.

Workflow examples

Average, a simple model-run diagnostic procedure to compute the average value of a climate variable (such as

Figure 4. Extreme value analysis plotting using an array of visualization tools, such as DV3D (upper left), VisIt-R plot (lower left), CDAT plots (middle), Visit-R (upper right), and ParaView (lower right). Using intuitive drag-and-drop operations, scientists can create, modify, copy, rearrange, and compare visualizations.



Variables

01 tas (324, 768, 1152)
-- pr (7300, 64, 128)
-- TEMP (42, 2400, 3600)
-- clt (120, 46, 72)
-- a (2, 80, 97)
-- u (2, 80, 97)
-- v (2, 80, 97)

Calculator

```
 [[ 243.51560974  243.51182556
243.51303101 ...,  243.51519775
   243.51525879  243.51535034]
  [ 243.89750671  243.89183044
243.89442444 ...,  243.89958191
   243.89216614  243.896698  ]
  [ 244.12406921  244.12866211
244.12910461 ...,  244.10774231
   244.11012268  244.11984253]
  ...,
  [ 242.27856445  242.27638245
242.25920105 ...,  242.26036072
   242.26329041  242.27088928]
  [ 241.95555115  241.95774841
241.95440674 ...,  241.95632935
   241.95597839  241.95489502]
  [ 241.74142456  241.74090576
241.74139404 ...,  241.7411499
   241.74125671  241.74133301]]]
>>> u+v
>>> a = u + v
```

Enter CDAT command and press Return

temperature) and hashvar, a typical IPCC-related analysis, illustrate the overall UV-CDAT workflow—purpose, data and metadata, tools, and visual results.

Average. The average operation is useful in determining the average value of a variable in the input files over many months or many years, and seeing how the average varies by location. A possible application is to calculate the climatological annual cycle of rainfall or surface temperature.

From a list of netCDF files and a list of variables of interest, the operation computes the average value for each latitude and longitude point for each variable over all the input files. It then creates a new netCDF file that has the average value for each variable.

In the output file, variable X at coordinate (0,0) is the average value for all Xs over the input files at coordinate (0,0).

Hashvar. As its name implies, hashvar is a frequency hashing operation, which is useful in spotting monthly or yearly consistent trends in the input files. For example, suppose a scientist wants to analyze if a temperature trend is present from a list of netCDF files. Hashvar determines the minimum and maximum values at each latitude and longitude point across all the files, with the bucket sizes based on the minimum, maximum, and selected number of bins, as well as how often a given latitude and longitude point is within a bucket range for all the files.

Hashvar also creates new netCDF files; the number of files depends on the number of buckets with internal netCDF limits on the number of allowed variables. The new files have X new variables per variable of interest (where X is the number of bins) that show the frequency for each latitude and longitude point over the set of input files.

In the output files, variable var_3 at coordinate (0,0) is the number of occurrences of {bin size 2} through {bin size 3} of variable var at coordinate (0,0) in all the input files.

Our goal is to build and deliver an advanced application that can locally and remotely access large-scale data archives, provides provenance and workflow functionality, as well as provide high-performance parallel analysis and visualization capabilities to the desktop of a geoscientist. These tools can help scientists make informed decisions about how best to meet the energy needs of the nation and the world in the context of climate change. Over the remaining part of 2013, the UV-CDAT team will continue to collaborate with national and international government agencies, universities, and corporations to extend parallel software capabilities to meet the challenging needs of ultrascale multimodel climate simulation and observation data archives.

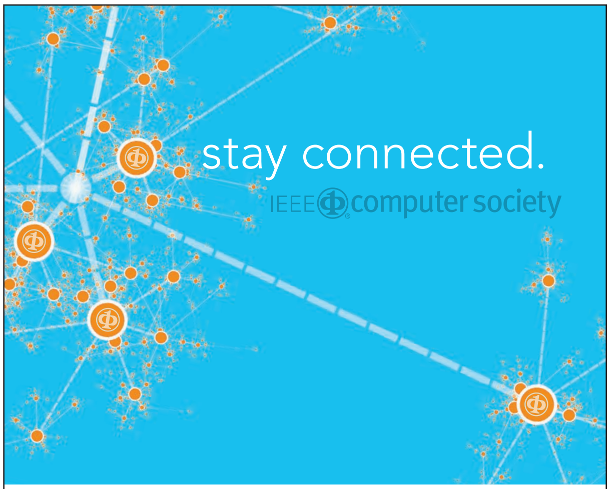
UV-CDAT is also useful in model development and testing. By analyzing 3D slices through time and space, scientists can isolate systematic errors in both forecast and climate simulations because they can simultaneously visu-

alize time and space. This unique view enables researchers to see model error growth, and allows first glimpses of model error attribution.

As geoscience datasets continue to expand in size and scope, there is a greater need to perform data analysis where the data is located (server-side analysis). UV-CDAT is therefore undergoing modifications to allow access to the DOE-sponsored Earth System Grid Federation infrastructure (www.esgf.org). With this modification, users can access petabyte archives and perform analysis and data reduction before moving the data to their site. Most important, the necessary remote operations will be routinely performed, thus freeing UV-CDAT users to concentrate on scientific diagnosis rather than on the mundane chores of data movement and manipulation.

Acknowledgments

This work is supported by the Director, Office of Science, Office of Biological and Environmental Research, of the US Department of Energy, under contracts DE-AC02-05CH11231 and DE-AC52-07NA27344 and by the National Aeronautics and Space Administration. This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the DOE Office of Science under Contract No. DE-AC05-00OR22725.



References

1. J.T. Overpeck et al., "Climate Data Challenges in the 21st Century," *Science*, vol. 331, no. 6018, 2011, pp. 700-702; doi: 10.1126/science.1197869.
2. J. Freire et al., "VisTrails: The Architecture of Open Source Applications," 2012; www.aosabook.org/en/vistrails.html.
3. J. Ahrens, B. Geveci, and C. Law, "ParaView: An End-User Tool for Large Data Visualization," *Energy*, vol. 836, 2005, pp. 717-732.
4. R. Drach, P. Dubois, and D. Williams, "Climate Data Management System, Version 5.0," 2007; www2-pcmdi.llnl.gov/cdat/manuals/cdms5.pdf.
5. C. Steed et al., "Practical Application of Parallel Coordinates for Climate Model Analysis," *Proc. Int'l Conf. Computational Science* (ICCS 12), Elsevier, 2012, pp. 877-886.
6. K. Martin and B. Hoffman, *Mastering CMake*, 4th ed., Kitware, 2008.

The UV-CDAT Project Team

Lawrence Livermore National Laboratory: *Dean N. Williams, UV-CDAT principal investigator; williams13@llnl.gov; Timo Bremer, bremer5@llnl.gov; Charles Doutriaux, doutriaux1@llnl.gov.*

Los Alamos National Laboratory: *John Patchett, patchett@lanl.gov; Sean Williams, seanw@lanl.gov.*

Oak Ridge National Laboratory: *Galen Shipman, gshipman@ornl.gov; Ross Miller, rgmiller@ornl.gov; David R. Pugmire, pugmire@ornl.gov; Brian Smith, smithbe@ornl.gov; Chad Steed, steedca@ornl.gov.*

Lawrence Berkeley National Laboratory: *E. Wes Bethel, ewbethel@lbl.gov; Hank Childs, hchilds@lbl.gov; Harinarayan Krishnan, hkrishnan@lbl.gov; Prabhat, prabhat@lbl.gov; Michael Wehner, MFWehner@lbl.gov.*

Polytechnic Institute of New York University: *Claudio T. Silva, csilva@nyu.edu; Emanuele Santos, emanuele@lia.ufc.br; David Koop, dkoop@poly.edu; Tommy Ellqvist, tommy.ellqvist@yahoo.se; Jorge Poco, jpocom@nyu.edu.*

Kitware: *Berk Geveci, berk.geveci@kitware.com; Aashish Chaudhary, aashish.chaudhary@kitware.com; Andy Bauer, andy.bauer@kitware.com.*

Tech-X: *Alexander Pletzer, pletzer@txcorp.com; Dave Kindig, kindig@txcorp.com.*

NASA Goddard Space Flight Center: *Gerald L. Potter, gerald.potter@nasa.gov; Thomas P. Maxwell, thomas.maxwell@nasa.gov.*