The Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT): Progress Report: January 1 through December 31, 2012

Dean N. Williams, williams13@llnl.gov
(Principle Investigator for UV-CDAT.)
Dr. Timo Bremer, bremer5@llnl.gov
(Computer scientist at LLNL, leading analysis and visualization of scientific data and applications.)
Charles Doutriaux, doutriaux1@llnl.gov
(Computer scientist at LLNL developing diagnostics and visualizations for the climate community.)
Lawrence Livermore National Laboratory (LLNL)
P.O. Box 808
Livermore (CA), 94550, U.S.A.
Phone: (925) 422-1100
Fax: (925) 422-7675

John Patchett, (Lead computer scientists at LANL in data and visualization.)
Sean Williams
Los Alamos National Laboratory (LANL)
P.O. Box 1663 MS B287
Los Alamos (NM), 87545, U.S.A.
Phone: (505) 665-1110
Fax: (505) 665-4939

Galen Shipman, gshipman@ornl.gov
(Leads ORNL's overarching strategy for data storage, management, and analysis for computational sciences.)
Ross Miller, rgmiller@ornl.gov
(Systems programmer for the National Center for Computational Sciences at ORNL.)
Daivd R. Pugmire, pugmire@ornl.gov
(Visualization task leader for the Oak Ridge Leadership Computing Facility (OLCF) at ORNL.)
Brian Smith, smithbe@ornl.gov
(Computer scientist investigating parallel algorithms and methods to improve big data analytics.)
Chad Steed, steedca@ornl.gov
(Computer Science Research Staff at ORNL whose research focuses on visual analytics and data mining.)
Oak Ridge National Laboratory (ORNL)
P.O. Box 2008 MS6164
Oak Ridge (TN), 37831-6164, U.S.A.
Phone: 865-576-2672
Fax: 865-574-6076

E. Wes Bethel, ewbethel@lbl.gov
(Principle Investigator for Visual Data Exploration and Analysis of Ultra-large Climate Data.)
Hank Childs, hchilds@lbl.gov
(Architect of VisIt—one of the most popular frameworks for data analysis and scientific visualization.)
Harinarayan Krishnan, (computer systems engineer at LBNL focused on providing software integration of the VisIt project within UV-CDAT.)
Prabhat, prabhat@lbl.gov
(Member of the Scientific Visualization group and the NERSC Analytics team at LBNL.)
Lawrence Berkeley National Laboratory (LBNL)
1 Cyclotron Road
Berkeley (CA), 94720, U.S.A.
Phone: (510) 495-2815
Fax:

Dr. Claudio T. Silva, csilva@nyu.edu
(Professor of computer science and engineering at NYU-Poly and Principle Investigator for VisTrails.)
Dr. Emanuele Santos, emanuele@lia.ufc.br
(Professor at the Federal University of Ceara in Brazil, teaching data science and visualization.)
Dr. David Koop, (NYU-Poly research assistant professor in the Department of Computer Science & Engineering.)
Tommy Ellqvist, tommy.ellqvist@yahoo.se
(Research assistant at NYU-Poly.)
Jorge Poco, jpocom@nyu.edu
(Ph.D. student at NYU-Poly.)
Polytechnic Institute of New York University (NYU-Poly)
6 Metrotech Pl, Brooklyn NY, 11201, U.S.A.
Phone: (718) 260-4093
Fax: (718) 260-3609

Berk Geveci, berk.geveci@kitware.com
(Director of Scientific Computing at Kitware and a leading developer of ParaView and VTK.)
Aashish Chaudhary, aashish.chaudhary@kitware.com
Dr. Andy Bauer, andy.bauer@kitware.com
(Researcher in the area of enabling technologies for large-scale PDE-based numerical simulations.)
Kitware, Inc.
28 Corporate Drive
Clifton Park (NY), 12065, U.S.A.
Phone: (518) 371-3971
Fax: (518) 371-4573

Alexander Pletzer, pletzer@txcorp.com
(Tech-X research scientist active in scientific programing, data analysis, modeling, and visualization.)
Dave Kindig, kindig@txcorp.com
(MA in Geography from the University of Colorado and currently working as a researcher at Tech-X.)
Tech-X Corporation
5621 Arapahoe Avenue Suite A
Boulder (CO), 80303, U.S.A.
Phone: (303) 448-0727
Fax: (303) 448-7756

Dr. Gerald L. Potter, gerald.potter@nasa.gov
(Analyst and data consultant at the NASA Center for Climate Simulation.)
Dr. Thomas P. Maxwell, thomas.maxwell@nasa.gov
(Lead scientist for the data analysis and visualization program at NASA Center for Climate Simulation.)
National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC)
Greenbelt (MD), 20771, U.S.A.
Phone: (301) 286-7810
Fax: (301) 286-1634

To support interactive visualization and analysis of complex, large-scale climate data sets, UV-CDAT integrates a powerful set of scientific computing libraries and applications to foster more efficient knowledge discovery. Connected through a provenance framework, the UV-CDAT components can be loosely coupled for fast integration or tightly coupled for greater functionality and communication with other components. This framework addresses many challenges in interactive visual analysis of distributed large-scale data for the climate community.

Keywords: Visualization, climate analysis, visual analytics, provenance.

I. Introduction: Background and History

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. Many speculate, the climate data deluge will continue to grow to unprecedented level of hundreds of exabytes for worldwide climate data holdings by 2020 [11]. Such explosive growth presents many challenges and unprecedented opportunity for the next round of scientific breakthroughs. In light of these challenges and opportunities, the Ultra-scale Climate Data Analysis Tools (UV-CDAT) address the visualization and analysis requirements of today's climate research institutions. In close collaboration with domain climate scientists, the first official software release of UV-CDAT was delivered to stakeholders in the climate community. Funded by the Department of Energy's (DOE's) Office of Biological and Environmental Research, in direct support of its climate science mission, the integrated, cross-institutional effort of computational and science teams consists of a consortium of four DOE national laboratories (Lawrence Berkeley [LBNL], Lawrence Livermore [LLNL], Los Alamos [LANL], and Oak Ridge [ORNL]); two universities (Polytechnic Institute of New York University [NYU-Poly] and the Univeristy of Utah; the National Aeronautics and Space Adminsitration (NASA) at Goddard Space Flight Center (GSFC); and two private companies (Kitware and Tech-X). To advance scientific analysis and visualization, the UV-CDAT team designed a framework that integrates several disparate technologies under one infrastructure (shown in Figure 1). United by standard common protocols and application programming interfaces (APIs), UV-CDAT integrates more than 40 different software libraries and components, of which the primary software stack comprises Climate Data Analysis Tools, VisTrails DV3D, ParaView, VisIt, EDEN, and R.

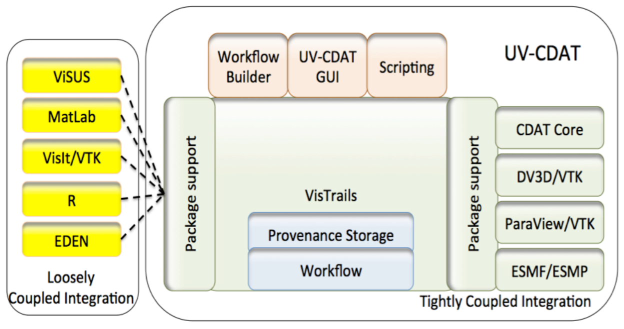


Figure 1. The UV-CDAT architecture shows the framework components, either integrated tightly or loosely coupled. From this design, other packages can be joined seamlessly.

UV-CDAT's Python-based framework enables visual data exploration and analysis capabilities of large-scale data sets. Its capabilities include parallel streaming statistics, analysis and visualization pipelines, optimized parallel input/output (I/O), remote interactive execution, workflow capabilities, and automatic data provenance processing and capturing. The novel graphical user interface (GUI) includes workflow data analysis and visualization construction tools as well as the ability to easily add custom functionality. These features are enhanced by the VisTrails provenance tool, the R statistical analysis tool, and state-of-the art visualization (DV3D, ParaView, EDEN, and VisIt), all of which are brought together within a Qt-based GUI. The architecture of UV-CDAT is presented with use cases illustrating the new capabilities of UV-CDAT in the areas of visualization, regridding, and statistical analysis.

Led by LLNL, this breakthrough development effort is coordinated nationally. Its primary goal is to build an ultra-scale data analysis and visualization system empowering scientists to engage in new and exciting data exchanges that may ultimately lead to breakthrough climate-science discoveries. The team has achieved their first goals:

- Official release of the UV-CDAT system version 1.2.
- Address projected scientific needs for data analysis and visualization.
- Extend UV-CDAT to support state-of-the-art regridding by interfacing to the Earth System Modeling Framework (ESMF) and LibCF libraries.
- Support climate model evaluation activities for DOE's climate applications and projects, such as the Intergovernmental Panel on Climate Change (IPCC) assessment report and Climate Science for a Sustainable Energy Future (CSSEF).

Over the years, the climate community has seen its models evolve toward higher numerical fidelity, which has caused data sets and archives to grow substantially, resulting in terabyte or more data sets and multi-petabyte archives. To this end, UV-CDAT is providing high-level solutions to address data- and climate-related issues as they pertain to analysis and visualization, such as:

- Problems with "big data" analysis (analytics).
- The need for reproducibility.

- Pushing ensemble, uncertainty quantification, and metrics computation to new boundaries.
- Heterogeneous data sources (simulations, observations, and reanalysis).
- Data analysis that cuts across multiple disciplinary domains.
- An overall architecture for incorporting existing and future software components.

Expanding UV-CDAT's community of developers and users facilitates the goal of continuing to adapt and meet the diverse scientific and computational challenges faced by the climate scientist. A primary motivation is to develop and use existing advanced software to disseminate and diagnose multi-model climate and observational data vital to understanding climate change. This interconnection of disparate software into a seamless infrastructure enables scientists to handle and analyze ever-increasing amounts of data and enhances their research by eliminating the need to write numerous large proprietary programs.

II. Basic Data, Metadata, and Grids

Data are critical to any research, and data formats play an integral role in the consolidation of geoscience information. In climate researach, data consist of two parts: the actual data resulting from model simulations, instruments, or observations, and the metadata that describe the data (e.g., how the data was generated, what the data represent, what is to be done with the data, how to use the data). Most collections of model runs, observations, and analysis files provide a uniform data access interface to such conventional formats as netCDF, HDF, GRIB, GRIB2, PP, and others. In the climate modeling simulation community and more recently the observation community, more groups are opting to store their data in the network Common Data Form (netCDF). The community has also selected a de facto methodology for defining metadata known as the Climate and Forecast (CF) metadata convention. Combining the netCDF–CF conventions makes it possible for other geoscience data sets to be compared and displayed together with very little effort on the part of the scientists. The netCDF–CF metadata conventions enables users of data from different sources to decide which quantities are comparable and facilitates building applications such as UV-CDAT with powerful extraction, regridding, and display capabilities. Adoption of these conventions has been facilated by the Climate Model Output Rewritter (CMOR) included in UV-CDAT, which makes it easy to produce properly formatted data.

III. The Analysis Processs

A. Pre-processing and regridding

A user survey recently revealed that regridding, i.e., the ability to interpolate data from one grid to another, is among the most widely used features in CDAT. In UV-CDAT, we extended this feature to support curvilinear grids. Ocean and atmospheric models often rely on curvilinear longitude–latitude grids in order to overcome numerical stability issues at the North and South Poles. Examples of curvilinear grids are the displaced/rotated pole grid and the tripolar grid, which are used by some ocean models to remove the North Pole singularity from the grid. The block-structured, cubed-sphere grid used by some atmospheric models is another example of a curvilinear grid with no singularity at the poles. Regridding Earth data presents a unique set of challenges.

First, the data may have missing or invalid values (e.g., ocean data values that fall on land). Second, users often demand that the total mass, energy, etc., be preserved after regridding. This conservative interpolation is the method of choice for cell-centered data but can be significantly more numerically intensive than nodal. We have addressed these challenges by leveraging multiple, existing, state-of-the-art interpolation libraries and by designing a single Python regridding interface supporting multiple interpolation tools (ESMF [4], LibCF [5], …) and methods (currently linear nodal, quadratic nodal, and conservative). (See Figure 2.) Depending on the type of grid (rectilinear or curvilinear) and the type of data (nodal or cell), the interface will automatically select the tool and method that is most appropriate for the task.

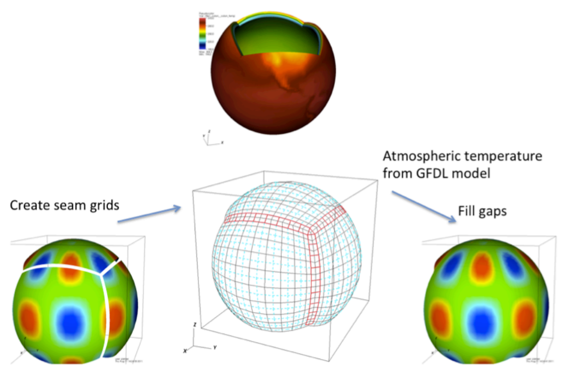


Figure 2. UV-CDAT supports mosaic grids using a number of well-established grid packages, such as: SCRIP, Gridspec, and ESMF. These standards comply with the needs established by CMIP. Shown are the supported cubed sphere and tripolar mosaic grids.

Regridding is accessible three different ways within the UV-CDAT GUI: Using the “REGRID” button on the calculator, the PCMDITool->Regrid menu item, or the Python command line interface. In all cases, two variables must be selected. The UV-CDAT interface regrids the first variable selected onto the second variable's grid. The last method allows the highest level of control because the user may select the tool and method as well as define how the periodicity of the data is handled and/or how the coordinate system is used for regridding, in the case of the ESMF tool. Selecting the coordinate system allows a longitude–latitude input grid to be converted to a Cartesian grid, thus avoiding singularities at the poles.

B. Exploratory data analysis and hypothesis generation

Another important aspect of UV-CDAT is its ability to provide users with the means to quickly explore massive amounts of data. This step is crucial for forming new hypotheses as well as verifying simulation data. Through the direct link to ParaView, VisIt, and DV3D, a scientist can now leverage three of the most import toolkits for visual data exploration from a single familiar interface (shown in Figure 3). UV-CDAT thus provides all traditional visualizations, such as slicing, volume rendering, isosurfacing etc., as well as the ability to explore long time series and to create animation sequences.

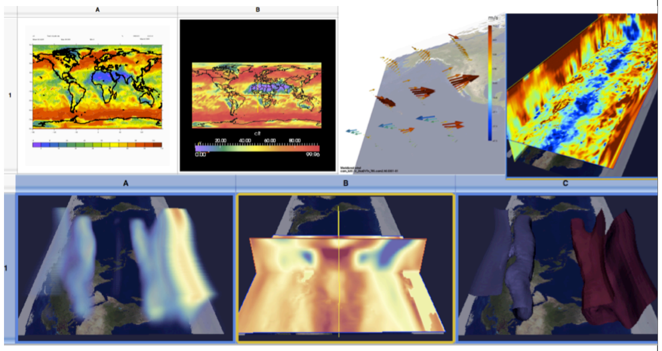


Figure 3. The UV-CDAT framework supports many 2D and 3D visualization techniques. The images shown here were accomplished with CDAT and DV3D visualization libraries.

C. Parallel processing

With climate models continuously improving numerical fidelity through increased resolution, there are cases where the memory footprint is too large for the data to reside on a single processor. On most platforms, this limit is somewhere around 10-km resolution for a single time step, global, 3D variable. To help users handle such memory-greedy processing and other numerically intensive operations, we have extended the behavior of the Climate Data Mangement System (CDMS) arrays in CDAT to allow remote memory access (RMA) within the scripting UV-CDAT environment. The implemented functionality supports remote data access via a a get method, which takes the remote processing rank and a tuple that uniquely represents a slice of the data to be fetched. This is implemented in Python using the mpi4py [6] module, and we rely on recent one-sided communication enhancements to the MPI-2 standard [7] for a concise implementation of distributed array functionality that works in any number of dimensions and for multiple data types. Although simple, the RMA implementation is more flexible than one based on point-to-point send/receive calls; the process that exports data need not know which process to send data to and each process can access data residing on any other processor.

D. Provenance

UV-CDAT is built on top of a provenance-enabled workflow system, and all its functionality is integrated into UV-CDAT. This means that during the analysis process, provenance information is automatically captured, making it possible to reproduce and share results and reducing the amount of effort to manage scripts and data files. Each analysis process has a corresponding workflow that is updated when the analysis changes. For example, when a parameter is changed or a new intermediate step is introduced. The updates are done incrementally so all versions of the analysis are kept in the provenance. To illustrate this, Figure 4a contains the workflow automatically generated for regridding a variable and plotting it using Boxfill plot type from the VCS library. The generated plot is displayed in Figure 4b.

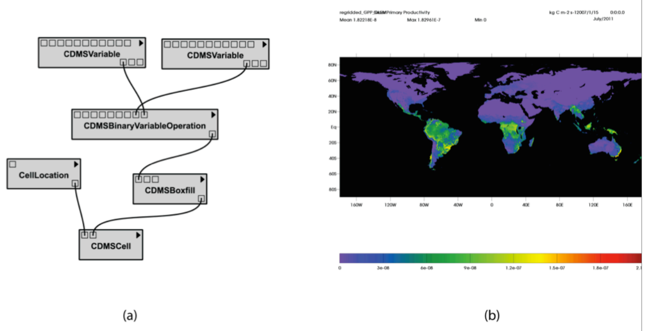


Figure 4. Example showing a UV-CDAT workflow (a) and the resulting plot (b).

E. Overall Workflow

The interaction between VisTrails and the UV-CDAT GUI is depicted in the diagram of Figure 5. As the user interacts with the UV-CDAT GUI by clicking buttons or dragging variables and plot types, a series of operations is generated and sent to VisTrails. These operations are converted into workflow operations, for example, module creation and parameter changes that are captured as provenance. VisTrails then notifies the system to update the plot and the GUI, if necessary.

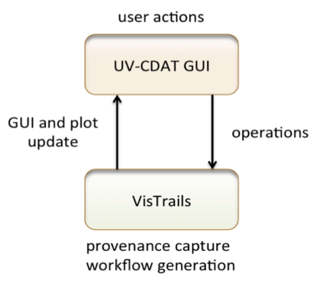


Figure 5. From the user's perspective, the diagram shows the UV-CDAT GUI and VisTrails workflow interactions.

IV. Community Tools and Environments

A. Quick inspection (e.g., CMake, CTest, CDash)

Developing large-scale software requires a rigorous software process to ensure quality systems. A widely used process is based on the open source tools such as: CMake, CTest, and CDash. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefile and workspaces. CTest is a testing tool distributed as a part of CMake. It could be used to automate code update, configure, build, and test. CDash is an open source web-based software-testing server. It aggregates, analyzes, and displays the result of software testing processes submitted from clients.

The software process for UV-CDAT consists of four major parts as shows in Figure 6: (1) a repository for data, documentation and code (Git is used for UV-CDAT's software repository); (2) a cross-platform build (using CMake); (3) a dashboard for collecting the results of testing (using CDash); and (4) finally the UV-CDAT developers that create, develop, and maintain the software. UV-CDAT software process is supportive of agile development methods, and is motivated by test-driven development approaches. Similar process is in use by thousands of software systems and has scaled to tens of millions of lines of code.

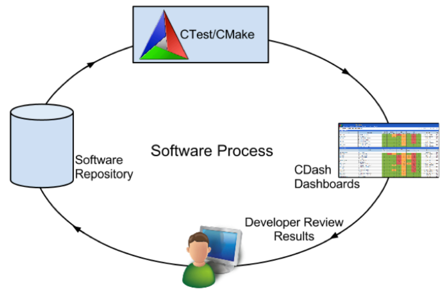


Figure 6. There is a four-step software process for vetting UV-CDAT's code base.

B. Community analysis and visualization tools and environments (e.g., R, CDAT, VisTrails, etc.)

VisTrails is an open-source system that supports data exploration and visualization. VisTrails allows the specification of computational processes that integrate existing applications, loosely-coupled resources, and libraries. A distinguishing feature of VisTrails is its provenance infrastructure. VisTrails captures and maintains a detailed history of the steps followed and data derived in the course of an exploratory task. It maintains provenance of data products (e.g., visualizations and plots) and the workflows that derive these products as well as their executions. VisTrails also provides a package mechanism to allow developers to expose their libraries (written in any language) to UV-CDAT by a thin Python interface through a set of VisTrails modules. This infrastructure makes it simple for users to integrate tools and libraries as well as to quickly prototype new functions.

C. 3D Tools (e.g., DV3D, ParaView, ViSUS, VisIt, R)

1) DV3D

DV3D is a VisTrails package of high-level modules for UV-CDAT that provides user-friendly workflow interfaces for advanced visualization and analysis of climate data at a level appropriate for scientists. This project is motivated by the observation that climate scientists can greatly benefit from advanced visualization methods, but they rarely use these tools because the existing interfaces are too complex. DV3D's simple GUI interface is designed to be readily used by busy scientists who have little time to invest in learning complex new frameworks. The application builds on Kitware's Visualization ToolKit (VTK), an open-source, object-oriented library, for visualization and analysis. DV3D provides the high-level interfaces, tools, and application integrations required to make the analysis and visualization power of VTK readily accessible to users without exposing details such as actors, cameras, renderers, and transfer functions. It can run as a desktop application or distributed over a set of nodes for hyperwall or distributed visualization applications.

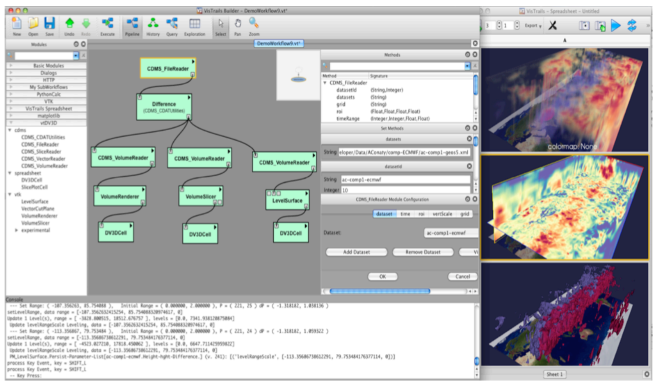


Figure 7. A DV3D workflow with spreadsheet output.

The 3D visualizations in Figures 7 were produced by DV3D. Its features include:

- Multiple interactive plot types, including volume rendering, volume slicing, isosurfaces, 3D hovmoller, parallel coordinates, and several vector field viewers (glyph slicer, glyph volume, and streamline slicer).
- A VisTrails workflow interface with custom GUIs for each module.
- A rich selection of interactive query, browse, navigation, and configuration options.
- Integration with the VisTrails spreadsheet, featuring multiple synchronized plots for desktop or hyperwall.
- Transparent and automated VisTrails provenance collection with provenance management interfaces.
- Active and passive VTK stereo vision support.
- Distributed task–parallel analysis pipelines.
- Seamless integration with CDAT's CDMS and other climate data analysis tools.

The DV3D package is composed of a set of VisTrails modules. These modules can be selected from a palette and linked to create custom data processing and visualization workflows using the VisTrails workflow builder. Alternately, the UV-CDAT GUI exposes a list of prebuilt DV3D workflows in its plot view. In this case, the user drags and drops a workflow onto each cell of the spreadsheet and configures the inputs to each workflow by dragging and dropping CDMS variables from the variable view. Figure 7 displays a typical DV3D workflow as it would appear in the VisTrails workflow builder interface of UV-CDAT and the resulting plots in the spreadsheet.

DV3D visualization modules encapsulates complex VTK pipelines with numerous supporting objects such as actors, cameras, renderers, interaction observers, data mappers, and transfer functions. Each branch of a DV3D workflow terminates in a DV3D cell module, which represents a cell in the UV-CDAT spreadsheet. The cell module features include a configurable base map, navigation controls, active and passive stereo vision toggle commands, onscreen data set and variable labels, a pick operation display, and variable and legend/colormap toggle commands. Cells in the spreadsheet can be individually activated or deactivated by selection, and configuration commands are automatically propagated to all active cells.

Each DV3D module offers a custom GUI interface (accessible from the VistTrails workflow builder), which enables the configuration of workflow parameters. The DV3D spreadsheet cells also offer a range of interactive click-and-drag operations that facilitate the configuration of colormaps, transfer functions, and other display and execution options. For example, clicking a button and then dragging in a spreadsheet cell that displays a DV3D volume rendering plot executes a leveling operation that controls the shape of the plot's opacity transfer function. The plot changes interactively as the user drags the mouse around the cell.

All configuration operations are saved as VisTrails provenance. The provenance trail makes it easy for the user to revert back to an earlier configuration of the workflow at any stage of development. It also maintains a record of all workflow construction and configuration operations that contribute to the current visualization. The user can maintain multiple developmental branches of a single workflow and easily switch back and forth between them.

2) *ParaView*

ParaView [8] is an open-source, multi-platform data analysis and visualization tool for interactive visualization of data on local or remote locations. The ParaView framework solves the large data visualization problem by using several approaches, such as parallel processing, client/server separation, and render server/data server separation. These approaches enable ParaView framework to run in stand-alone or client server mode. In the standalone mode, data processing and rendering are performed locally on a system, whereas in the client server mode most of the data processing and rendering are performed on the server, with only the geometry or rendered images sent to the client. The UV-CDAT framework tightly integrates ParaView to take advantage of its large data visualization capability. Within the UV-CDAT framework, a user can create a ParaView pipeline by creating a workflow using the UV-CDAT GUI.

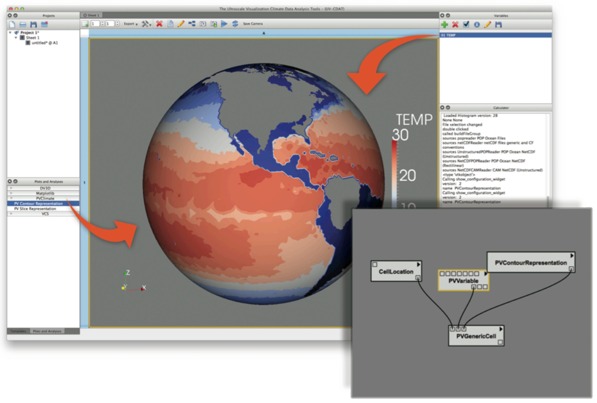


Figure 8. Steps to create a ParaView workflow in UV-CDAT framework.

As shown in Figure 8, an appropriate workflow is created and executed when a user drops a ParaView representation and a variable (a PVVariable instance in this case) on a spreadsheet cell. Integration of ParaView within UV-CDAT allows an user to create multiple representations for a variable. For instance, for a variable, an user can create a contour and a slice represenation in the same view. This is possible because both of these representations share a single view between them. Each of the ParaView representations contains its own data pipeline and when executed, generates visualization in a view that is possibly shared between representations. ParaView can be run in a standalone or client server mode within the UV-CDAT framework. In its current state, a user connects to a ParaView server using the Python shell. Once connected, a user can browse the remote file system to select data sets for visualization purposes.

Work is in progress to support spatio-temporal parallelism within the UV-CDAT framework using ParaView.

3) *ViSUS: Streaming Visualization*

These visualization and analysis packages are comparatively large and full-featured applications aimed at providing a broad range of tools covering a variety of scientific disciplines. While they are powerful and highly general, such tools often become less effective in strongly resource-constrained environments. For example, dealing with large data sets can become cumbersome in the case of small-scale resources, i.e., a laptop or remotely via limited network bandwidth. To deal with these use cases, UV-CDAT is integrating a new complementary technology based on the ViSUS (Visualization Streams for Ultimate Scalability) framework [12]. At its core, ViSUS is focused on providing fast, multi-resolution, cache-oblivious access to extreme size data sets. Based on the concept of hierarchical space-filling curves, ViSUS provides a progressive and multi-resolution *stream* of data that drastically reduces the amount of file I/O necessary to extract, for example, a slice of data from a 3D data set. As a result, ViSUS has demonstrated interactive access to terabytes of simulation data on devices as small as a smartphone and remotely using low-bandwidth connections such as public WiFi hotspots. The ViSUS architecture consists of two components: A visualization client running under various GUI front ends including a web-browser and a light-weight server encapsulating the (remote) data access. With the the client integrated into UV-CDAT, a scientist can easily explore remote data sets directly from a personal desktop before committing to extensive data transfers or remote analysis efforts.

Additionally, the streaming nature of ViSUS's data pipeline enables some unique capabilities, such as on-the-fly processing. As noted, some of the most common climate data operations involve computing various temporal and/or spatial averages or simple statistics, such as standard deviations and variances. Given the extreme size of today's data sets, such computations require extensive resources in both disk space and time, which typically makes an extensive exploration infeasible. For example, given an ensemble of climate models, a single mean model might be created for analysis while means of different subsets may be equally of interest. ViSUS allows computation of such statistics or derived data on the fly as data are read from disk. An example is a multi-model average that is computed by simultaneously extracting data from all models and averaging the resulting data sets. Because file I/O is typically the dominant cost, such processing can be pipelined with little effect on the overall performance of the system. Furthermore, in a remote setting, the processing can be performed entirely on the server, resulting in the identical amount of data being transferred compared to the single model case.

In this manner, ViSUS allows a scientist to virtually explore, for example, a mean model using all standard visualization tools while never instantiating the actual mean model. One immediate advantage is that the user can easily add or remove models from the mean provide and thus explore the space of all possible combinations of models. Similarly, temporal averages and per-pixel descriptive statistics (variance, skewness, kurtosis, etc.) can be computed.

4) VisIt

VisIt is an open source, turnkey application for large scale simulated and experimental data sets. Its charter goes beyond just making pretty pictures; the application is an infrastructure for parallelized, general post-processing of extremely massive data sets. Target use cases include data exploration, comparative analysis, visual debugging, quantitative analysis, and presentation graphics. The basic design is a client-server model, where the server is parallelized. The client-server aspect allows for effective visualization in a remote setting, while the parallelization of the server allows for large data sets to be processed reasonably interactively. The tool has been used to visualize many large data sets, including a two hundred and sixteen billion data point structured grid, a one billion point particle simulation, and curvilinear, unstructured, and Adaptive Mesh Refinement (AMR) meshes with hundreds of millions to billions of elements. Within UV-CDAT, VisIt has a loosely coupled infrastructure, which means that the client components are wrapped and integrated within UV-CDAT whereas the server component is executed separately. This mode enables VisIt to execute climate analysis algorithms on machines that leverage distributed processing either locally or remotely.

The client side integration within UV-CDAT allows VisIt to offer all its rendering and analysis capabilities. These include over twenty-one plots (ways to render data), forty-two operators (ways to manipulate data), eighty-five file format readers, fifty queries (ways to extract quantitative information), and one hundred expressions (ways to create derived quantities). Further, a plugin capability allows for dynamic incorporation of new plot, operator, and database modules. For UV-CDAT, VisIt provides a custom user interface and exposes its entire set of Qt-based widgets to the main PyQt UV-CDAT application. This allows commonly used features such as connecting to remote machines, editing visualization results, and executing large distributed runs on high performance clusters to be executed with ease.

As part of the UV-CDAT project several new climate specific operations were added to VisIt. Two specific operations include computing Peaks-over-Threshold and Extreme Value Analysis. Both these operations utilize GNU-R scripts at their core and utilize the new VTK-R Bridge to interface with VisIt as well as UV-CDAT. For example, the Extreme Value Analysis operation is used to estimate past and future changes in extreme precipitation (and other climate variables) using model output and observations. Figure 9 shows a computation of the Extreme Value Analysis operation using VisIt-R on the upper left and a example rendering of temperatures using VisIt on the lower right along with plots of DV3D, CDAT, and ParaView.

5) R

R is a package for statistical computing that is widely used within the climate community. By incorporating this package into VisIt and UV-CDAT we are able to leverage many statistical analysis and algorithms that are at the heart of many climate analysis work. Currently, custom R scripts are used within VisIt to compute several climate related operations. We are actively working on making R procedures available to the rest of UV-CDAT.

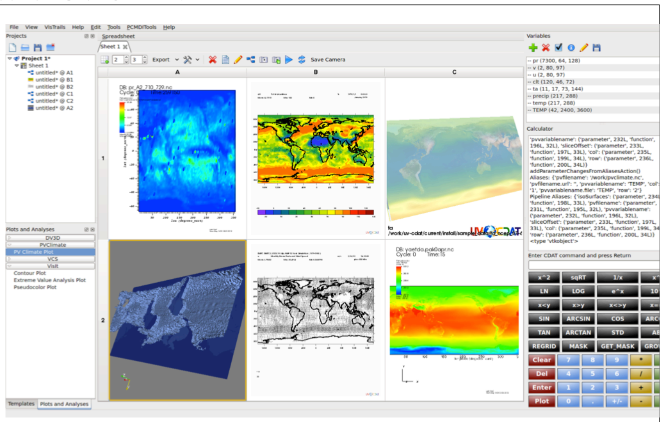


Figure 9. Extreme value analysis plotting using an array of visualization tools, such as: VisIt-R (upper left), ParaView plot (lower left), CDAT plot (middle), DV3D (upper right), and VisIt (lower right).

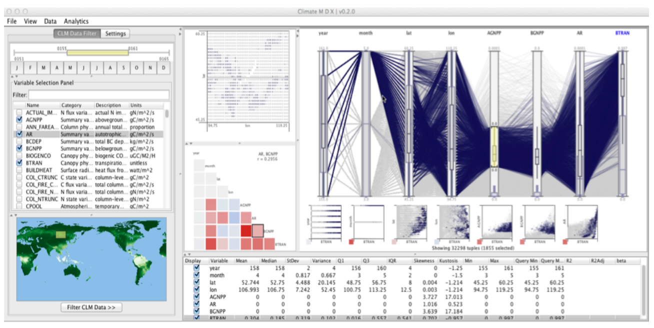
D. Graphical user interface

The UV-CDAT GUI, the main window for UV-CDAT, is shown in Figure 9. It is based on the notion of a spreadsheet (middle) or a resizable grid in which each cell contains a visualization. By using intuitive drag-and-drop operations, visualizations can be created, modified, copied, rearranged, and compared. Spreadsheets maintain their provenance and can be saved and reloaded. These visualizations can be used for data exploration and decision-making, while at the same time being completely customizable and reproducible.

Around the spreadsheet are the tools for building visualizations, as shown in Figure 9. The project view (top left) allows one to group spreadsheets into projects and to name visualizations and spreadsheets. The plot view (bottom left) allows for the use and customization of available plot types. The variable view (top right) allows data variables to be edited. The bottom right contains a variable editor widget, making editing a variable similar to using a pocket calculator. Using the UV-CDAT GUI, creating a visualization can be as simple as dragging a variable to a spreadsheet cell and then dragging a plot type to the same cell.

E. Ensemble Data analysis ENvironment (EDEN)

Two of the most challenging tasks in exploring multi-dimensional climatic data sets are identifying and quantifying the associations among a complex set of interrelated variables. The scientist often has some understanding of expected relationships, but unexpected discoveries using conventional approaches are difficult at best. With the Ensemble Data analysis ENvironment (EDEN) tool (see Figure 10), interactive information visualization techniques are combined with automated statistical analytics in a visual analytics framework that effectively guides the scientist to the most significant associations in the data. EDEN is built around a central parallel coordinates plot [8] that is combined with scatterplots and a correlation matrix view. EDEN capitalizes on the perceptual benefits of multiple coordinated views so that brushings and selections of the data in one view are propagated appropriately to the other views. A more detailed description of EDEN is provided in a prior publication [9].



	Name	Category	Description	Units
☐	ACTUAL_IM...	N flux varia...	actual N im...	gN/m^2/s
✓	AGNPP	Summary va...	abovegroun...	gC/m^2/s
☐	ANN_FAREA...	Column phy...	annual total...	proportion
✓	AR	Summary va...	autotrophic...	gC/m^2/s
☐	BCDEP	Summary va...	total BC dep...	kg/m^2/s
✓	BGNPP	Summary va...	belowgroun...	gC/m^2/s
☐	BIOGENCO	Canopy phy...	biogenic CO...	uGC/M2/H
✓	BTRAN	Canopy phy...	transpiratio...	unitless
☐	BUILDHEAT	Surface radi...	heat flux fro...	watt/m^2
☐	COL_CTRUNC	C state vari...	column-leve...	gC/m^2
☐	COL_FIRE_C...	C flux varia...	total column...	gC/m^2/s
☐	COL_FIRE_N...	N flux varia...	total column...	gN/m^2/s
☐	COL_NTRUNC	N state vari...	column-leve...	gN/m^2
☐	CPOOL	Atmospheri...	temporary...	gC/m^2

Display	Variable	Mean	Median	StDev	Variance	Q1	Q3	IQR	Skewness	Kustosis	Min	Max	Query Min	Query M...	R2	R2Adj	beta
✓	year	158	158	2	4	156	160	4	0	-1.25	155	161	155	161			
✓	month	4	4	0.817	0.667	3	5	2	0	-1.5	3	5	3	5			
✓	lat	52.744	52.75	4.488	20.145	48.75	56.75	8	0.004	-1.214	45.25	60.25	45.25	60.25			
✓	lon	106.993	106.75	7.242	52.45	100.75	113.25	12.5	0.003	-1.214	94.75	119.25	94.75	119.25			
✓	AGNPP	0	0	0	0	0	0	0	3.727	17.013	0	0	0	0			
✓	AR	0	0	0	0	0	0	0	1.016	0.523	0	0	0	0			
✓	BGNPP	0	0	0	0	0	0	0	3.639	17.184	0	0	0	0			
✓	BTRAN	0.304	0.185	0.319	0.102	0.016	0.557	0.541	0.702	-0.957	0	0.997	0	0.997			

Figure 10. The EDEN visual analytics framework provides an interactive parallel coordinates display that has been extended with several linked visualization and augmented with information produced by automated statistical analytics. The visualization panel consists of a geographic point plot (upper left), a correlation matrix view (lower left), and a parallel coordinates/scatterplot view (right).

With its filter panel (see left panel of Figure 10), EDEN enables selecting a portion of Fourth Community Land Model (CLM4) data sets based on time (years and months of interest), a geographical region, and a set of variables to retrieve. After these parameters are selected, EDEN retrieves the data from a user-defined CLM4 simulation data set and populates the visualization panel on the right. EDEN also functions in a more general form, without the CLM4 filter panel, reading data in the form of Comma Separated Value (CSV) files.

The parallel coordinate axes are enhanced with visual cues that guide the scientist's exploration of the information space. Each parallel coordinate axis is augmented with graphical encodings of several key descriptive statistics (e.g., mean / median and interquantile / standard deviation ranges) to summarize the variable distributions. This variability information is then encoded in the boxes rendered in each axis interior. The wide boxes represent statistics for all the axis samples while the narrow boxes capture the samples that are currently queried. Portions of an axis can be queried by dragging a box over the axis range using mouse gestures.

EDEN also displays small scatterplots below the parallel coordinates panel when an axis is highlighted. The scatterplots are created by assigning the highlighted variable as the y-axis and the variable directly above the scatterplot as the x-axis. The scatterplots provide a means to visually confirm correlations between variables, particularly when nonlinear relationships exist. Scatterplots can be selected and viewed in a separate window to engage in more detailed analysis.

EDEN enables a visual form of correlation mining to judge the strength of relationships between pairs of variables. As the user interacts with the system, the correlation information is updated in real-time and used to augment the display. For each possible pairing of axes, the system automatically calculates the Pearson product-moment correlation coefficient, yielding a correlation matrix that is displayed graphically at the lower-left pane of the visualization panel. The graphical correlation indicators are also displayed in the lower left portion of the scatterplots as color-coded boxes. In each box, the color encodes the correlation coefficient between the x and y axes of the scatterplot. The correlation indicator boxes are shaded blue for negative correlations and red for positive. The stronger the correlation, the more saturated the color so the stronger correlations are more visually salient.

F. *Emerging integrated collaborative environments*

Anticipating the adoption of UV-CDAT as the de-facto analysis tool first by the climate community and subsequently by other research domains, the UV-CDAT developers realized that eventually they will not be able to answer all calls for enhancement. As a result, the UV-CDAT Reusable Analysis and Diagnosis (U-ReAD) Framework is being set up to stimulate that sharing of diagnosis and analysis among scientists. The ambitious goal of this framework is to enable scientists to "upgrade" their code into a UV-CDAT plugin by relying mainly on documentation.

The advantages for the scientist are many: Integration into UV-CDAT means greater exposure, better documented code means easier adoption by others, and integration into UV-CDAT means provenance and workflows are available "for free."

Adoption of this framework will also allow for better collaboration among scientists and greater exposure for young scientists. This project should also result in positive feedback to UV-CDAT. As the number of "plugins" for UV-CDAT increases so will UV-CDAT adoption, fueling even more development via the U-ReAD Framework.

V. Examples

Mini-case-studies where we describe the overall workflow (i.e., what purpose, what data and metadata, tool or tools, and visual results). These will range from simple model-run diagnostics, to typical IPCC-related analyses, to 3D renderings of various scenarios.

A. *Average –*

The map-average program takes a list of netCDF files and a list of variables of interest. It then computes the average value for each latitude and lonitude point for each of the variables of interest over all of the input files. It 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). This can be useful for 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.

The above also applies for standard deviation.

B. *Hashvar –*

The frequency hashing program takes a list of netCDF files and a list of variables of interest plus a number of bins to create. It determines the minimum and maximum values for each variable at each latitude and longitude point across all the files,the bucket sizes based on the minimum, maximum, and number of bins, and the frequency that a given latitude and longitude point is within a bucket range for all of the files. One or more new netCDF files are created (depending on the number of buckets with internal netCDF limits on the number of allowed variables). The new files have {number of bins} new variables per variable of interest that show the frequency for each latitude and longitude point over the set of input files. In the output file(s), 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 of the input files. This can be useful for spotting trends in the input files that are consistent month-to-month or year-to-year.

VI. Future Challenges and Directions

Our goal is to build and deliver an advanced application (UV-CDAT) that can locally and remotely access large-scale data archives, offer provenance and workflow functionality, and provide high-performance parallel analysis and visualization capabilities to the desktop of a geoscientist who will apply these tools to make informed decisions on meeting the energy needs of the nation and the world in light of climate change consequences. Over the coming year, the UV-CDAT team of developers will continue to collaborate with national and international government agencies, universities, and corporations to extend parallel software capabilities to meet the challenging needs of ultra-scale multi-model climate simulation and observation data archives.

To meet project deliverables, the following tasks are proposed:

(1) Solve solutions to cross-cutting parallel use cases for a rich set of diverse and complex projects, such as the Fifth Coupled Model Intercomparison Project (CMIP5), NASA satellite data, and reanalysis data sets.

(2) Incorporate regridding, reprojections, and aggregation for simulation and observational data sets.

(3) Implement needed model diagnostics and observational tools.

(4) Add comparative visualization and analysis.

(5) Implement and explore the use of streaming visualization for largescale computing.

Another use of UV-CDAT is model development and testing. Using 3D slicing through time and space it is possible to isolate systematic errors in both forecast and climate simulations because the user can visualize time and space at the same time. This unique view enables the researcher to see model errors grow and allows first glimpses of model error attribution.

As geoscience data sets continue to expand in size and scope, the necessity for performing data analysis where the data is co-located (i.e., server-side analysis) is becoming increasingly apparent. UV-CDAT is therefore undergoing modifications to allow access to the DOE-sponsored Earth System Grid Federation (ESGF) infrastructure. This modification will allow users to not only access petabyte archives, but also to peform analysis and data reduction before moving the data to their site. Most importantly, 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 manipulation.

Acknowledgments

The development and operation of UV-CDAT is supported by the U.S. Department of Energy | Office of Science Biological and Environmental Research and the NASA High-End Computing (HEC) Program through the NASA Center for Climate Simulation (NCCS) at Goddard Space Flight Center.

Prepared by LLNL under Contract DE-AC52-07NA27344.

References

[1] UV-CDAT home page: http://uvcdat.llnl.gov/
[2] UV-CDAT wiki: http://uvcdat.llnl.gov/wiki/
[3] UV-CDAT source code repository:
[4] ESMF library: http://www.earthsystemmodeling.org/
[5] LibCF library: http://www.unidata.ucar.edu/software/libcf/
[6] Mpi4py: http://mpi4py.scipy.org/
[7] MPI-2: http://www.mpi-forum.org/docs/docs.html
[8] Inselberg, A., The Plane with Parallel Coordinates. *The Visual Computer*, 1 (4), (1985), pp. 69-91.
[9] Steed, C. A., Shipman, G., Thornton, P., Ricciuto, D., Erickson, D., Branstetter, M. Practical Application of Parallel Coordinates for Climate Model Analysis. In *Proceedings of the International Conference on Computational Science*, pp. 877-886.
[10] Ahrens, J., Geveci, B. & Law, C. ParaView: An End-User Tool for Large Data Visualization. *Energy* **836**, 717-732 (2005).
[11] CKD Workshop, Climate Knowledge Discovery Workshop, March 2011, DKRZ, Hamburg, Germany, https://redmine.dkrz.de/collaboration/projects/ckd-workshop/wiki/CKD_2011_Hamburg.
[12] ViSUS home page: http://visus.us/