The Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT): Data Analysis and Visualization for Geoscience Data

Dean Williams, Charles Doutriaux, John Patchett, Sean Williams, Galen Shipman, Ross Miller, Chad Steed, Harinarayan Krishnan, Claudio Silva, Aashish Chaudhary, Peer-Timo Bremer, David Pugmire, E. Wes Bethel, Hank Childs, Prabhat, Berk Geveci, Andrew Bauer, Alexander Pletzer, Jorge Poco, Tommy Ellqvist, Emanuele Santos, Gerald Potter, Brian Smith, Thomas Maxwell, David Kindig, and David Koop

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

May, 2013

Acknowledgment

This work was supported by the Director, Office of Science, Office of Biological and Environmental Research and the Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Legal Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.

The Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT): Data Analysis and Visualization for Geoscience Data

Dean N. Williams, williams13@llnl.gov
(Principal 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, patchett@lanl.gov
(Lead computer scientists at LANL in data and visualization.)
Sean Williams, seanw@lanl.gov
(Applied computer scientist and visualization expert at LANL.)
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.)
Dr. David 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

Dr. E. Wes Bethel, ewbethel@lbl.gov
(Principal Investigator for Visual Data Exploration and Analysis of Ultra-large Climate Data.)
Dr. Hank Childs, hchilds@lbl.gov
(Computer Systems Engineer at LBNL and Architect of VisIt—one of the most popular frameworks for data analysis and scientific visualization.)
Dr. 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: (510) 486-5812

Dr. Claudio T. Silva, csilva@nyu.edu
(Professor of computer science and engineering at NYU-Poly and Principal 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, dkoop@poly.edu
(Research assistant professor in the Department of Computer Science & Engineering at NYU-Poly.)
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 the interactive visual analysis of distributed large-scale data for the climate community.

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

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 that the climate data deluge will continue to grow to unprecedented levels of hundreds of exabytes for worldwide climate data holdings by 2020 [1]. Such explosive growth is a double-edged sword presenting both challenges and opportunity for the next round of scientific breakthroughs. We have developed the Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT) [2] to address the visualization and analysis needs of today's big data climate analysis in close collaboration with domain experts in climate science. UV-CDAT provides high-level solutions to address data and climate related issues as they pertain to analysis and visualization, such as:

- Problems with "big data" analytics;
- The need for reproducibility;
- Pushing ensemble analysis, uncertainty quantification, and metrics computation to new boundaries;
- Heterogeneous data sources (simulations, observations, and re-analysis);
- Data analysis that cuts across multiple disciplinary domains; and
- An overall architecture for incorporting existing and future software components.

The integrated, cross-institutional effort, comprised of computational and domain scientists 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, we designed a Python-based framework that integrates several disparate technologies under one infrastructure (see Figure 1). United by standard common protocols and application programming interfaces (APIs), UV-CDAT integrates more than 40 different software components. The primary software stack of the various components comprises Climate Data Analysis Tools, VisTrails, DV3D, and ParaView.

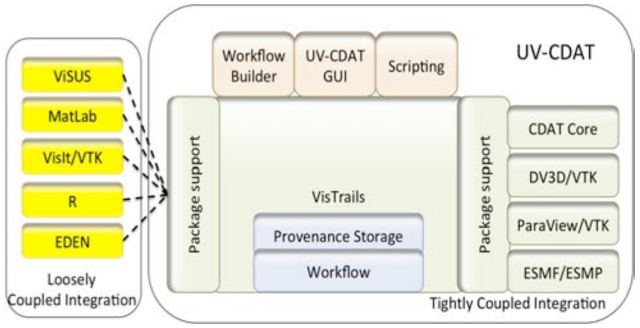

Figure 1. The UV-CDAT architecture shows the framework components, either integrated tightly or loosely coupled. From this design, other packages can be joined seamlessly.

The primary goal of this nationally coordinated effort 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. To date, our team has achieved the following major objectives:

- Released the official UV-CDAT system version 1.2.
- Addressed projected scientific needs for data analysis and visualization.
- Extended UV-CDAT to support the latest regridding by interfacing to the Earth System Modeling Framework (ESMF) [3] and LibCF libraries.
- Supporting on-going 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) [4].

Expanding UV-CDAT's community of developers and users facilitates our goal of evolving, and meets the diverse scientific and computational challenges faced by the climate scientist. Our 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 will enable scientists to handle and analyze ever-increasing amounts of data, and enhances their research by eliminating the need to master numerous different frameworks.

UV-CDAT brings to bear a number of capabilities that are intended to directly address climate scientists' needs. The strengths of this framework include parallel streaming statistics, optimized parallel input/output (I/O), remote interactive execution, workflow capabilities, and automatic data provenance capture. In addition to the ability to intuitively add custom functionality, the user interface includes tools for workflow analysis and visualization construction. We augment these capabilities with other features such as linkage to the R statistical analysis environment and enhanced visualization tools (DV3D, ParaView, EDEN, and VisIt), all of which are integrated under a Python/Qt-based architecture. In this paper, we describe the UV-CDAT architecture with use cases illustrating the new capabilities of UV-CDAT in the areas of visualization, regridding, and statistical analysis.

II. Basic Data, Metadata, and Grids

Data is critical to any research, and data formats play an integral role in the consolidation of geoscience information. In climate research, data consists of two parts: 1) the actual data resulting from model simulations, instruments, or observations, and 2) the metadata that describes the data (e.g., how the data was generated, what the data represents, 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 conventional formats, such 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. When working with data from different sources, the netCDF–CF metadata conventions enable users 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 is possible through inclusion of the Climate Model Output Rewriter (CMOR) included in UV-CDAT, which makes it easy to produce properly formatted data.

III. The Analysis Process

A. 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 interpolation. We have addressed these challenges by leveraging multiple existing interpolation libraries, and by designing a single Python regridding interface supporting multiple interpolation tools (ESMF[5], SCRIP [6]) and methods (currently linear nodal, quadratic nodal, and conservative). 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.

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 CDAT, ParaView, VisIt, and DV3D, a scientist can now leverage four important toolkits for visual data exploration from a single common interface (shown in Figure 2 and 4). 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 to create animation sequences. UV-CDAT uses a spreadsheet paradigm that allows for combinations of different plots, including 2D and 3D plots.

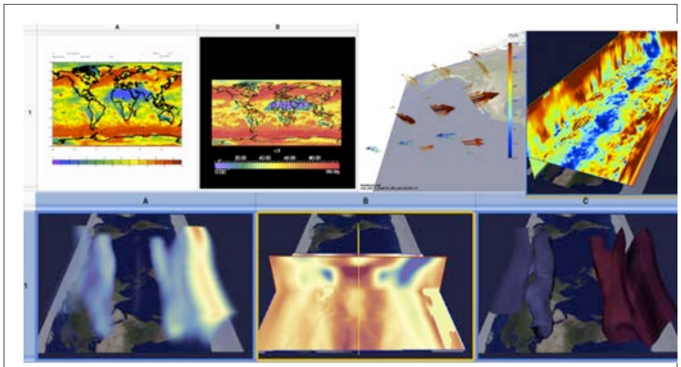

Figure 2. The UV-CDAT framework supports many 2D and 3D visualization techniques. The images shown in this collage are products of the CDAT and DV3D visualization libraries.

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 will accommodate loading and processing one 3D variable at 10km resolution at a single time step. 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) [7] arrays in CDAT to allow remote memory access (RMA) within the scripting UV-CDAT environment. The new functionality supports remote data access via 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 [8] module, and we rely on recent one-sided communication enhancements to the MPI-2 standard 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 an Open Source provenance-enabled workflow system called VisTrails [9]. During the analysis process, VisTrails automatically captures provenance information, making it possible to reproduce and share results, and thereby, 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 (e.g., 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 3a contains the workflow automatically generated for regridding a variable and plotting it using the Boxfill plot type from the CDAT library. The resulting plot is shown in Figure 3b.

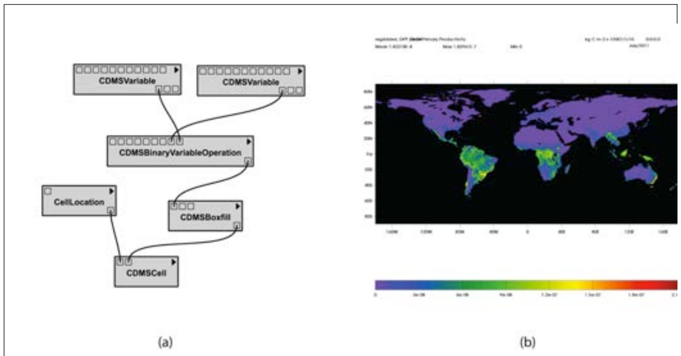

Figure 3. Example showing a UV-CDAT workflow (a) and the resulting plot (b).

E. Workflow Generation

As the user interacts with the UV-CDAT Graphical User Interface (GUI) by clicking buttons or dragging variables and plot types, a series of operation events are generated and processed by the VisTrails API. These events are converted into workflow operations (e.g., module creation and parameter changes) that are captured as provenance. VisTrails then notifies the system to update the plots and the GUI, as necessary. It is also possible to directly edit workflows, and to create new plots by using the workflow builder.

IV. Community Tools and Environments

A. Quick inspection

Developing large-scale software requires a rigorous software process to ensure quality systems. A widely used process is based on the Open Source tools CMake, CTest, and Cdash [10]. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces. CTest is a testing tool distributed as a part of CMake. It can be used to automate code update, configuration, build, and test operations. CDash is an Open Source web-based software-testing server. It aggregates, analyzes, and displays the results of software testing processes submitted from clients.

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

B. Community visualization and analysis components

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 the provenance of data products and the workflows that derive these products, as well as their executions. VisTrails also provides a package mechanism, allowing developers to expose their libraries (written in any language) to UV-CDAT using a thin Python interface encapsulated by 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. DV3D

DV3D is a VisTrails package of high-level modules for UV-CDAT that provides user-friendly workflow interfaces for advanced climate data visualization and analysis. DV3D provides the interfaces, tools, and application integrations required to make the analysis and visualization power of VTK [11] 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 applications.

The DV3D package offers scientists a set of coordinated interactive 3D plot types that provide insightful views of data sets. Each DV3D plot type offers a unique perspective by highlighting particular features of the data. Multiple plots can be combined synergistically to facilitate an understanding of the natural processes underlying the data. The plot types include:

- *Volume slice*. The volume slice plot provides a set of slice planes that can be interactively dragged over data sets. This tool allows scientist to very quickly and easily browse the 3D structure of data sets, compare variables in 3D, and probe data values.

- *Volume render*. The volume render plot maps variable values within a data volume, varying with opacity and color. It enables scientists to create an overview of the topology of the data, revealing complex 3D structures at a glance.

- *Hovmoller volume slice*. 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 extensive data processing and analysis functionality. DV3D expands the scientists' toolbox by incorporating a suite of rich new exploratory visualization and analysis methods for addressing the complexity of climate data sets.

D. ParaView

ParaView [12] is an Open Source, multi-platform data analysis and visualization tool for interactive visualization of data on local or from 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 the ParaView framework to run in stand-alone or client-server mode. In the standalone mode, data processing and rendering are performed locally on the client, 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 so as 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.

Integration of ParaView within UV-CDAT allows a user to create multiple visualizations of the same variable. For instance, a user can create a contour and a slice represenation of a single variable in the same shared view. 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.

E. ViSUS: Streaming Visualization

Dealing with large data sets can become cumbersome in the case of small-scale resources. To deal with these use cases, UV-CDAT is integrating a new complementary technology based on the ViSUS (Visualization Streams for Ultimate Scalability) framework [13]. 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 information (e.g., 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, while 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 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.

F. VisIt

VisIt [14] is an Open Source, turnkey application for visualizing and analyzing large-scale simulation 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 architecture 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 216 billion data point structured grid and a one billion point particle simulation data set, as well as 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 while 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.

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 4 shows a computation of the Extreme Value Analysis operation using VisIt-R on the upper left, and an example rendering of temperatures using VisIt on the lower right along with plots of DV3D, CDAT, and ParaView.

G. R

R [15] 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 algorithms that are at the heart of much climate analysis work. Currently, VisIt uses custom R scripts to compute several climate-related operations. We are actively working on making R procedures available to the rest of UV-CDAT.

H. EDEN

The Exploratory Data analysis ENvironment (EDEN) [16]. fulfills the need for a visual data mining capability in UV-CDAT. EDEN blends interactive information visualization techniques with automated statistical analytics to effectively guide the scientist to the most significant relationships. EDEN is built upon a set of coordinated views with central parallel coordinate visualization. EDEN has been developed collaboratively with climate researchers on the CSSEF project.

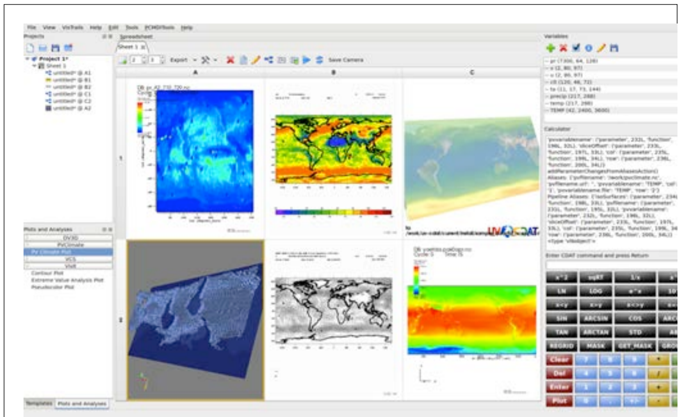

Figure 4. Extreme value analysis plotting using an array of visualization tools, such as: VisIt-R (upper left), ParaView plot (lower left), CDAT plots (middle), DV3D (upper right), and VisIt (lower right).

I. Graphical user interface

The UV-CDAT GUI is shown in Figure 4. It is based on the notion of a VisTrails visualization 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.

In Figure 4, located around the spreadsheet are the tools for building visualizations. The project panel (top left) allows one to 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. At the bottom right, a calculator widget can be used to derive new variables using computations. To create a visualization in UV-CDAT, a user drags a variable from the variable panel and a plot type from the plot list to a spreadsheet cell.

V. Examples

This section describes mini-case-studies illustrating the overall UV-CDAT workflow (i.e., what purpose, what data and metadata, tool or tools, and visual results). These include a simple model-run diagnostic procedure and a typical IPCC-related analysis.

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 longitude 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. This process has also been applied using a standard deviation operator.

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, that provides provenance and workflow functionality, and that provides 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.

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 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) [17] infrastructure. This modification will allow users to access petabyte archives and 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 movement and manipulation.

Acknowledgments

This work is supported by the Director, Office of Science, Office of Biological and Environmental Research, of the U.S. Department of Energy, under Contract Numberss DE-AC02-05CH11231 and DE-AC52-07NA27344 and by the National Aeronautics and Space Administration.

References

[1] Overpeck, J.T., G. A. Meehl, S. Bony, and D. R. Easterling, 2011: Climate Data Challenges in the 21st Century. Science, 331, 700-702, doi: 10.1126/science.1197869.

[2] UV-CDAT home page: http://www.uv-cdat.org/

[3] ESMF home page: http://www.earthsystemmodeling.org/

[4] DOE's Office of Biological and Environmental Research (BER) climate modeling project home page: http://www.climatemodeling.science.energy.gov/projects/

[5] ESMF home page: http://www.earthsystemmodeling.org/

[6] SCRIP home page: http://climate.lanl.gov/Software/SCRIP/

[7] R. Drach, P. Dubois, and D. Williams, 2007: Climate Data Management System, version 5.0, http://www2-pcmdi.llnl.gov/cdat/manuals/cdms5.pdf

[8] MPI for Python home page (mpi4py): http://mpi4py.scipy.org/

[9] "VisTrails", Juliana Freire, David Koop, Emanuele Santos, Carlos Scheidegger, Claudio Silva, and Huy T. Vo, The Architecture of Open Source Applications, 2012. http://www.aosabook.org/en/vistrails.html

[10] Martin, Ken, and Bill Hoffman. Mastering CMake 4th Edition. Kitware, Inc., 2008. ISBN-13: 978-1930934221

[11] VTK home page: http://www.vtk.org/

[12] Ahrens, J., Geveci, B. & Law, C. ParaView: An End-User Tool for Large Data Visualization. Energy 836, 717-732 (2005).

[13] ViSUS home page: http://visus.us/

[14] VisIt home page: https://wci.llnl.gov/codes/visit/home.html/

[15] R home page: http://www.r-project.org/

[16] 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.

[17] ESGF home page: http://www.esgf.org/