LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-655463

Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) Final Report

D. N. Williams

June 5, 2014

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) Final Report

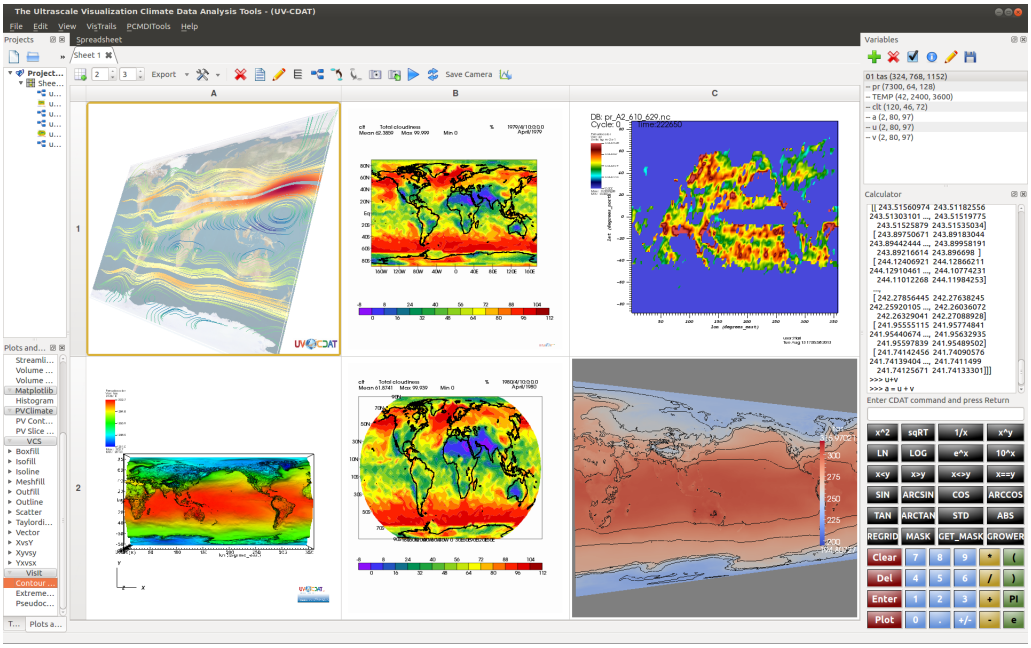
October 1, 2013, through April 30, 2014

Principal Investigator

Dean N. Williams[3]

The UV-CDAT Team

Andrew Bauer[1], Aashish Chaudhary[1], Berk Geveci[1],
Curtis Canada[2], Phil Jones[2], Boonthanome Nouanesengsy[2],
David Bader[3], Timo Bremer[3], Charles Doutriaux[3], Matthew Harris[3], Elo Leung[3], Renata McCoy[3],
Thomas Maxwell[4], Gerald Potter[4],
Cecelia DeLuca[5], Ryan O'Kuinghttons[5], Robert Oehmke[5],
Ben Burnett[6], Aritra Dasgupta[6], Tommy Ellqvist[6], David Koop[6], Emanuele Marques[6], Jorge Poco[6], Rémi Rampin[6],
Claudio Silva[6], Huy Vo[6],
John Harney[7], David Pugmire[7], Galen Shipman[7], Brian Smith[7], Chad Steed[7],
David Kindig[8], Alexander Pletzer[8]



Extreme-scale analysis and visualization for complex Earth system science data

[1] Kitware, Inc.
[2] Los Alamos National Laboratory (LANL)
[3] Lawrence Livermore National Laboratory (LLNL)
[4] National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC)
[5] National Oceanic and Atmospheric Administration (NOAA)
[6] New York University (NYU)
[7] Oak Ridge National Laboratory (ORNL)
[8] Tech-X, Inc.

Abstract

A partnership across government, academic, and private sectors has created a novel system that enables climate researchers to solve current and emerging data analysis and visualization challenges. The Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT) software project utilizes the Python application programming interface (API) combined with C/C++/Fortran implementations for performance-critical software that offers the best compromise between "scalability" and “ease-of-use.” The UV-CDAT system is highly extensible and customizable for high-performance interactive and batch visualization and analysis for climate science and other disciplines of geosciences. For complex, climate data-intensive computing, UV-CDAT’s inclusive framework supports Message Passing Interface (MPI) parallelism as well as taskfarming and other forms of parallelism. More specifically, the UV-CDAT framework supports the execution of Python scripts running in parallel using the MPI executable commands and leverages Department of Energy (DOE)-funded general-purpose, scalable parallel visualization tools such as ParaView and VisIt. This is the first system to be successfully designed in this way and with these features. The climate community leverages these tools and others, in support of a parallel client-server paradigm, allowing extreme-scale, server-side computing for maximum possible speed-up.

The project team worked closely with current and proposed scientific programs within DOE’s Office of Biological and Environmental Research (BER) to advance the development of state-of-the-art tools in support of BER’s science mission. The primary goal of the consortium—three DOE laboratories (LANL, LLNL, and ORNL), NASA, NOAA, two universities (New York University and University of Utah), and two private companies (Kitware and Tech-X)—was to explore and develop software and workflow applications needed to:

- Integrate DOE’s climate modeling and measurements archives.
- Develop infrastructure for national and international simulation and observation data comparisons.
- Deploy a wide range of climate data visualization, diagnostic, model metric, and analysis tools with familiar interfaces for very large, high-resolution climate data sets to meet the growing demands of this data-rich community.

The screen shot of UV-CDAT’s thick-client application on the title page shows a collage of disparate visualization products, all joined seamlessly under our framework.

Introduction

Increasingly large climate model simulations are enhancing our understanding of the processes and causes of anthropogenic climate change, thanks to very large public investments in high-performance computing at national and international institutions. Various climate models implement mathematical approximations of nature in different ways, often based on differing computational grids. These complex, parallelized coupled system codes combine the integration of numerous complex sub-models (ocean, atmosphere, land, biosphere, sea ice, land ice, etc.) that represent components of the larger complex climate system. Climate scientists learn from these simulations by comparing modeled and observed data. A variety of grid schemes, and temporal and spatial resolutions make this task challenging even for small data sets. Recent advances in high-performance computing technologies are enabling the production, storage, and analysis of multi-petabyte output data sets.

As climate models become more complex and output data sets larger, the steps involving generation, movement, and analysis of model output severely tax serial data processing capabilities. Moreover, support for services that provide remote access to large-scale data is essential, as community-wide

analysis of model results becomes commonplace and as generalized diagnostics and analysis such as multi-model ensembles become available. Therefore, from the perspective of large-scale data processing, major efforts from the DOE, NASA, and NOAA are being devoted to:

- Enabling codes (and hardware and networks) to efficiently output simulation results to parallel disk systems.
- Developing analysis tools and workflow patterns to efficiently post-process large volumes of climate model output for scientific purposes.
- Improving data structures for parallel data processing of batch and interactive processing.

In addition, high-resolution models, ensemble analysis, derived data-product generation, and intercomparison of model results and observations require substantial data-storage infrastructure, in terms of both online data storage (high-performance parallel input/output [I/O] environments) and archival data storage. Managing these data sets—from generation to transformation and fusion to archiving—requires a robust infrastructure that supports multiple models and storage systems. Test-bed simulations, such as the DOE Accelerated Climate Modeling for Energy (ACME) project, require parallel I/O environments that provide a structured data model suitable for parallelizing and automating ensemble analysis and intercomparison of models and observational data sets. Observational data sets come in a variety of formats and use numerous metadata models. Supporting the data-storage requirements of these data sets—and data products derived from these data sets—requires decoupling the data format from data access and exploratory/parallel analysis.

To this end, UV-CDAT was designed to incorporate parallel streaming statistics, analysis and visualization pipelines, optimized parallel I/O, remote interactive execution, workflow capabilities, and automatic data provenance processing and capturing. UV-CDAT also offers a novel graphical user interface (GUI) and scripting capabilities for scientists that include 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 advancements in state-of-the-art visualization (DV3D, ParaView, and VisIt), all of which are brought together within a Qt-based GUI. The architecture of UV-CDAT was designed around use cases utilizing capabilities in the areas of visualization, regridding, and statistical analysis. These use cases, shown in **Figure 1**, can be found on UV-CDAT's Website: https://github.com/UV-CDAT/uvcdat/wiki/Use-Cases.

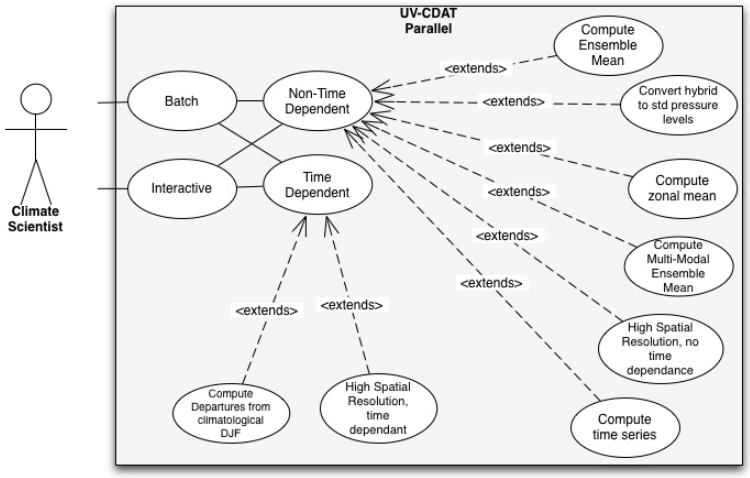


Figure 1. *Use case diagram for UV-CDAT depicting non-time-dependent and time-dependent parallel use cases running in batch or interactive mode. Many of the use cases used for UV-CDAT's development account for tasks commonly performed by climate model analysts that require explicit spatial, temporal, spatio-temporal and multi-attribute considerations.*

Project Design

UV-CDAT is an integrated framework that provides an end-to-end solution for management, analysis, and visualization of the ultrascale data sets generated for current and future BER climate data repositories and for the climate science community at large. UV-CDAT is based on a client–server architecture and is integrated within the Earth System Grid Federation (ESGF) framework, allowing UV-CDAT to take advantage of the advanced data management mechanisms of ESGF. In this way, UV-CDAT provides regridding, reprojection, and aggregation tools directly as a component of the ESGF data node, eliminating or substantially decreasing data movement. The UV-CDAT client provides a turnkey application for building complex data analysis and visualization workflows by interacting with one or more UV-CDAT servers. These workflows may use predefined components for data transformation and analysis, data collection from disparate data sources outside ESGF, visualization, as well as user-defined processing steps.

The UV-CDAT framework couples powerful software infrastructures through two primary means:

- **Tightly coupled integration** of the CDAT Core with the VisTrails/DV3D/VTK/ParaView/ESMF infrastructure to provide high-performance parallel streaming data analysis and visualization of massive climate data sets; and
- **Loosely coupled integration** to provide the flexibility to use tools such as VisIt, ViSUS, R, MatLab, and ParCat for data analysis and visualization as well as to apply customized data analysis applications within an integrated environment.

With both paradigms, UV-CDAT provides data provenance capture and mechanisms to support data analysis via the VisTrails infrastructure as shown in **Figure 2**.

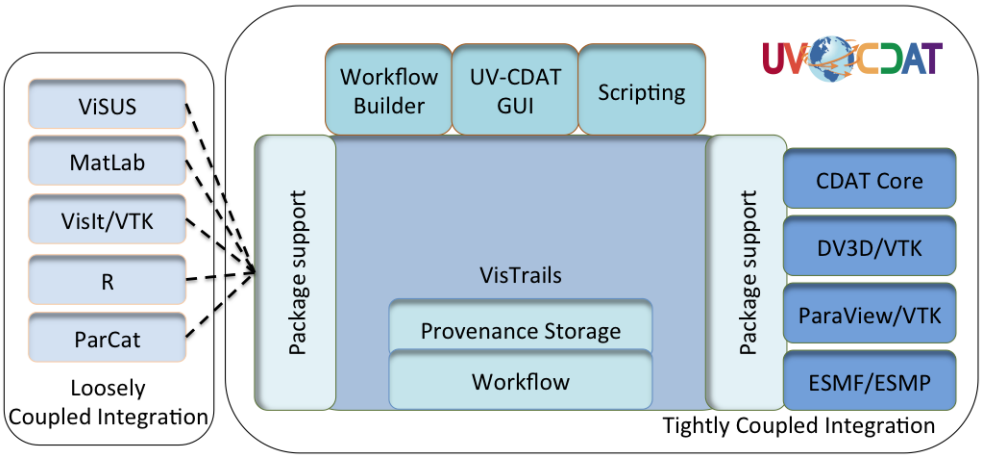


Figure 2. *The UV-CDAT architecture can integrate a large collection of components, either tightly or loosely coupled. From this design, other packages can be joined seamlessly.*

On the server side, UV-CDAT's Python-based API provides the ability to read data from local or remote sources, run analysis algorithms on local or remote computing resources in serial or parallel mode, and visualize result of algorithms output in a thick client (such as desktop GUI) or a smart client (such as Web browser). The desktop GUI uses the UV-CDAT Python API, whereas communication between the smart client and the Python framework uses the latest in Web technologies, such as Web-sockets and a restful API. In its current format, UV-CDAT on the Web server can use the computing horsepower provided by a cluster or a supercomputer. The ability to connect to other instances of UV-CDAT compute nodes is under development.

Our Web-based analysis and visualization system, named UV-CDAT Live, uses the traditional client–server architecture concept within the Web-based model, shown in **Figure 3**. It is more similar to the thick-client concept in that UV-CDAT Live smart clients are Internet-connected devices that allow a user's local applications to interact with server-based applications through the use of Web services. This process allows for more analysis and visualization interaction and software customization, which is similar to thick clients but without the hassle of software downloads and installation.

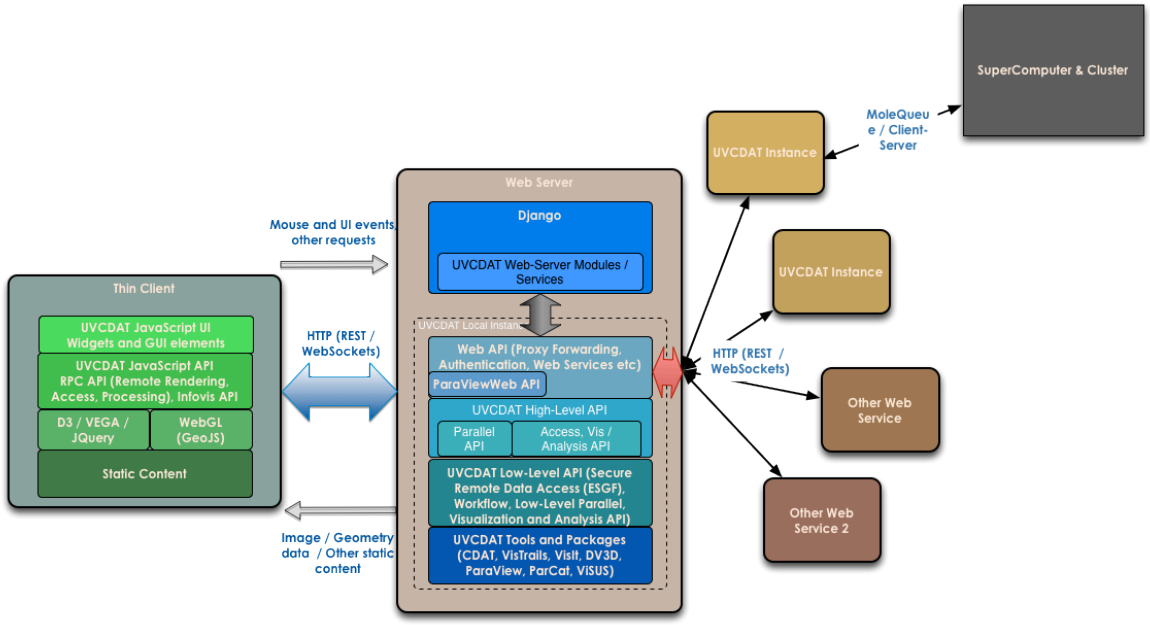


Figure 3. *An image of the UV-CDAT Live Web-based architecture for Web informatics and server-side analysis, diagnostics, model metric, and visualization.*

To support systematic maintenance of data provenance and to ensure reproducibility, UV-CDAT has designed a mechanism that provides reliable (and persistent) links between workflows and the data they generate. Flexible schemes were designed into the framework to allow application-specific provenance to be integrated into the provenance store. This feature helps users understand previous analyses as well as construct new ones; mechanisms for querying provenance are essential. The provenance information is stored as XML files and in a relational database. With the intuitive user interface, users can navigate workflow versions, undo changes but not lose any results, visually compare different workflows and their results, and examine the actions that led to a result.

Implementation of the Project Design

To achieve this design, CDAT, VisTrails, DV3D, ParaView, and VTK provide high-performance, parallel-streaming data analysis and visualization of massive climate data sets, and they are tightly coupled for greater system performance. UV-CDAT is also a highly flexible and extensible system. Using VisTrails's package mechanism, it is simple to loosely integrate (or incorporate) other tools, including VisIt, ViSUS, R, and MatLab, for data analysis and visualization without modifying the system core (see **Figure 2**). This package mechanism enables developers to expose their own libraries (written in any language) to the system by a thin—Web-based—Python interface through a set of VisTrails modules. Users are able to interact with the system in different ways: by using the UV-CDAT GUI, the VisTrails' workflow builder, through Python scripts, or via a Web browser.

The UV-CDAT software development team (and indeed the community) has accomplished something that has never before been attempted, much less completed, at this level of software engineering for the climate community: the integration of more than 70 disparate scientific software packages and libraries for large-scale data analysis and visualization. **Table 1** includes a list of the packages incorporated into UV-CDAT. It should be noted that the build system (i.e., using CMake, CDash, and CTest) enables users to select a subset of these tools for any particular UV-CDAT-based custom application.

Table 1. *UV-CDAT builds upon over 70 external software packages.*

basemap 1.0.5	pkgconfig 0.23.0	termcap 1.3.1	readline 6.2	libxml2 2.7.8
libxslt 1.1.26	CURL 7.22.0	YASM 1.2.0	ffmpeg 0.11.1	zlib 1.2.5
png 1.5.1	CDAT 1.3.1	jpeg v8c	tiff 3.9.4	pbmplus
Tlc/Tk 8.5.9	LAPACK/CLAPACK	freetype 2.4.10	Pixman 0.30.0	fontconfig 2.10.1
Cairo 1.10.2	uuid 1.6.2	udunits2 2.1.24	shapely 1.2.14	HDF5 1.8.8
NetCDF 4.3.0	Qt 4.7.2	jasper 1.900.1	g2clib 1.2.5	Python 2.7.4
VisIt 2.6.0	SIP 4.14.6	PyQt 4.10.1	PyOpenGL	Numpy 1.7.1
Pmw 1.3.2	CMOR 2.8.3	Matplotlib 1.2.0	GEOS 3.3.5	Libcf 1.0-beta11
Cython 0.16	MPI 1.6.4	ESMF 6.1.0	setuptools 0.6c11	gui_support
distribute 0.6.45	Pip 1.3.1	MyProxyClient 1.3.0	Numexpr 2.1	Lepl 5.1.3
Sphinx 1.2.b1	pyzmq 13.1.0	spyder 2.2.0	tornado 3.1	IPYTHON 0.13
Mpi4py 1.3	NetCDFPLUS 4.2.1.1	R 2.15.1	ParaView 3.11.1	Pyspharm 1.0.7
PyTables 2.4.0	SCIPY 0.12.0	scikits 0.12	VTK 5.9.0	VisTrails
DV3D	CDATLogger	WGET 1.12	gdal 1.9.1	docutiles 0.10
jinja2 2.7	pyopenssl 0.13	pygments 1.6	blaze	llvm
llvmpy	Django 1.5.1	Numba 0.9		

Software Code Releases

From October 1, 2010 through September 20, 2013, UV-CDAT was under development. Version 1.0 was officially released in early 2013. The next official full release of UV-CDAT (version 2.0) is due out in the middle of 2014. With the release of UV-CDAT, substantial analysis and visualization, particularly at global and continental scales, can be locally and remotely accessed from ultrascale globally federated data archives. UV-CDAT delivers high-performance parallel analysis and visualization capabilities to the desktops of multiple scientists, and it helps scientists make informed decisions on climate change consequences.

Software release dates are shown in **Table 2**. For more information, visit the UV-CDAT Website (http://uv-cdat.org) or the "Roadmap to Release" UV-CDAT GitHub wiki site (https://github.com/UV-CDAT/uvcdat/wiki/Roadmap-to-Release).

Table 2. *UV-CDAT's past and future software release schedule.*

Product	***Release Date***	***End of Engineering***	***End of Life/ End of Support***
UV-CDAT 2.0	July 2014	Planned	Planned
UV-CDAT 1.5.0	April 9, 2014	Current	Current
UV-CDAT 1.4.0	September 25, 2013	January 4, 2013	January 4, 2014
UV-CDAT 1.3.0	April 29, 2013	August 5, 2013	September 18, 2013
UV-CDAT 1.2.0	February 4, 2013	August 5, 2013	April 29, 2013
UV-CDAT 1.0.0	January 6, 2013	February 4, 2013	February 4, 2013
UV-CDAT Pre-release	May 9, 2012	May 20, 2012	June 29, 2012
UV-CDAT Beta	April 4, 2012	May 9, 2012	June 29, 2012
UV-CDAT Alpha	February 2, 2012	April 4, 2012	May 9, 2012

Progress and Deliverables During Transition Period

Below is the progress made over the past months on the transition period from UV-CDAT/CSSEF to the possible DOE Next-Generation ACME project. The image of the new UV-CDAT GUI and the UV-CDAT Live user interface are shown in **Figure 4**.

- Designed UV-CDAT diagnostics framework.
- Developed documentation of diagnostics API.
- Integrated diagnostics API into UV-CDAT thick client.
- Over 1000 diagnostics were generated for the Atmospheric Modeling Working Group (AMWG).
- Developed support for more complicated Land Modeling Working Group (LMWG) diagnostics.
- Designed and developed of connection to diagnostics with UV-CDAT thick client and Web informatics user interface smart client.
- Designed and developed the start of Web Informatics for UV-CDAT.
- Designed and developed the start of server-side analysis.
- Designed and developed the start of the ESGF connection and text search functionality Implemented ESGF login for data access.
- Designed and developed of a prototype of the HTML template for demonstrating of server-side capabilities.

- Designed and developed of a prototype of interactive VCS plot using server-side rendering.
- Designed and developed the start of exploratory analysis.

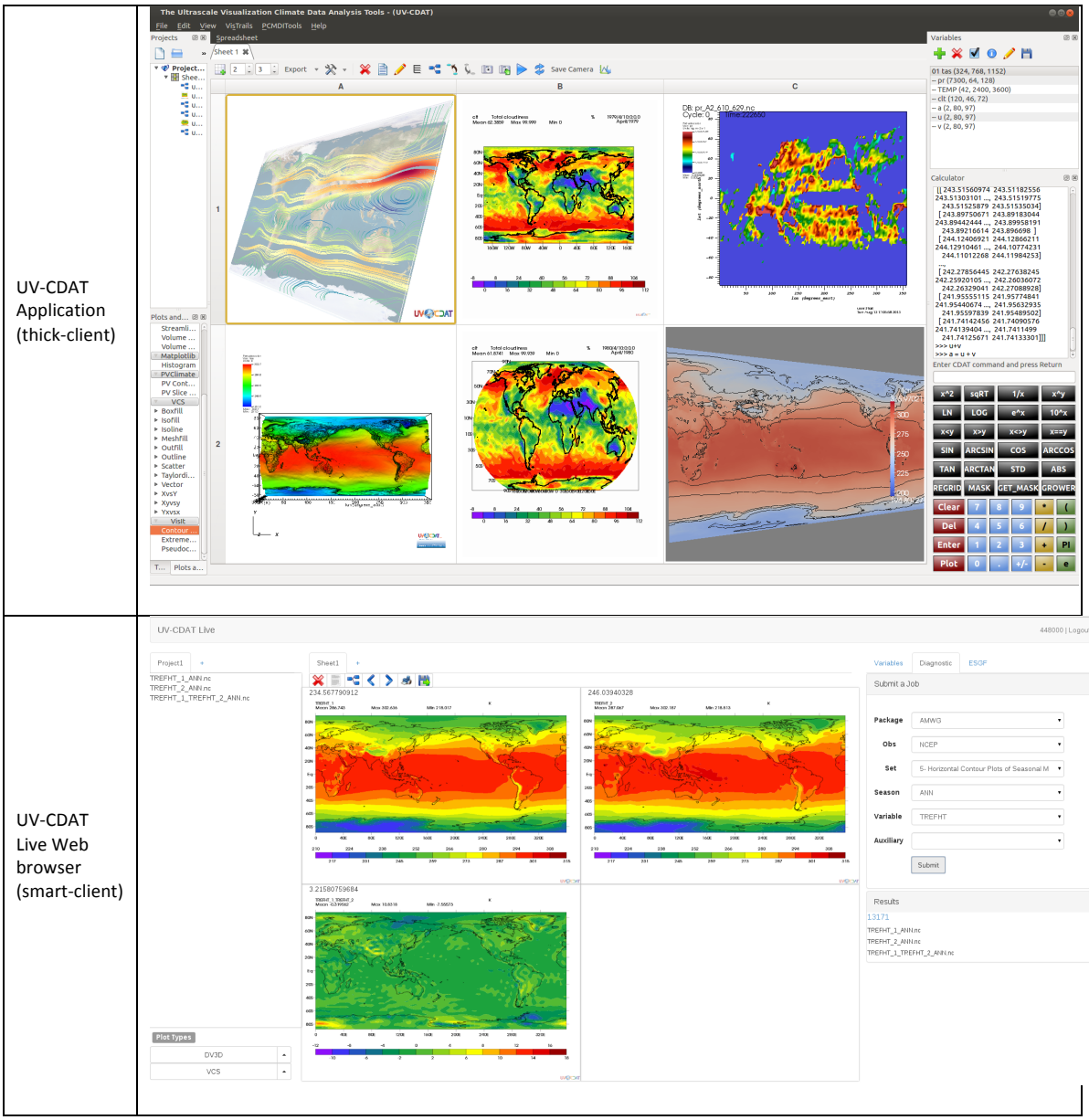


Figure 4. *Top image shows the UV-CDAT think-client desktop/laptop application displaying multiple analysis/visualization package output. The bottom image displays the UV-CDAT Live smart-client, run though the Firefox browser, displaying AMWG diagnostics.*

Future Plans and Discussion

In the future, UV-CDAT's development team will integrate future techniques and develop a general cyber infrastructure for processing big climate data for analysis and visualization purposes under the DOE BER ACME project. The integration and continued development of the parallel data infrastructure to support processing, analysis, and visualization of big climate data will focus on three core mechanisms:

1. The first mechanism uses UV-CDAT in batch execution in a "bag of tasks" model. Each compute node is assigned a distinct UV-CDAT workflow to execute. This workflow is beneficial for many common analysis tasks such as processing current-generation model diagnostics in which individual diagnostic functions can be assigned to a single UV-CDAT process. Parallelization is achieved by running multiple UV-CDAT processes, and hence multiple diagnostic functions, concurrently across a large number of compute nodes.
2. The second mechanism provides parallelization of analysis and data summarization tasks across larger-scale data sets in a batch post-processing mode. This mode uses the UV-CDAT ParCAT system that supports high-performance parallel I/O through libraries such as Parallel NetCDF and extremely efficient analysis operators developed using the MPI and native C routines. This parallel infrastructure is most suited for batch post-processing of very-large-scale climate data sets across large-scale compute resources.
3. The third mechanism provides parallelization of analysis and visualization tasks across larger-scale data sets for interactive and batch analysis. This mechanism relies on the client–server architecture of UV-CDAT coupled with ParaView/VisIt. Together, they provide parallel analysis across both temporal and spatial domains directly using the mature ParaView/VisIt analysis and visualization infrastructure while supporting the interactive analysis and visualization aspects of UV-CDAT.

Ensuring that the software developed for UV-CDAT will be sustainable after the end of project funding is an important consideration. By making our software available to the science community under open-source licenses, we will enhance its sustainability and invite community contributions via outreach, mailing lists, project Website (http://uv-cdat.org/) and wiki pages (https://github.com/UV-CDAT/uvcdat/wiki/). In addition, we will investigate other key technologies to incorporate into the software stack. Since our research is focused on running parallel software in large DOE high-performance compute environments (i.e., Leadership Computing Facilities (LCFs)), we will explore other models for running our parallel compute software. Scientists hosting data and running workflows on the DOE machines are already using those resources, as noted above. We will experiment in the next years of the project under ACME with hosting our proposed services on clusters using GPUs and CPUs, and with their combined use by scientists who prefer not to deploy the services on DOE LCF environments.

UV-CDAT's success can be measured by its expanding use. It is now integrated with the international ESGF system as a front-end access mechanism to acquire data for analysis and visualization and as a prototype back-end tool to reduce data sets and return visualization products. It is also expanding into other DOE- and NASA-funded projects as the cornerstone of interagency proposed projects. In particular, two future projects under DOE's Earth System Modeling effort aim to use UV-CDAT to deliver new capabilities that will further facilitate interactive and visual exploration and diagnostics of simulation and observational output. These projects share a joint vision for large-scale visualization and analysis of climate data and will work to organize and expand UV-CDAT's capabilities.

Community Outreach

Report	Dean N. Williams, et al., *3rd Annual Earth System Grid Federation and Ultrascale Visualization Climate Data Analysis Tools Face-to-Face Meeting Report*. Technical report, February 2013. [Online] http://aims-group.github.io/pdf/ESGF_UV-CDAT_Meeting_Report_Feb262014.pdf.
Report	Dean N. Williams, et al., *Ultrascale Visualization Climate Data Analysis Tools: Three-Year Comprehensive Report.* Technical report, October 2013. [Online] http://uv-cdat.org/media/pdf/three-year-comprehensive-report.pdf.
Paper	"Web-based Visual Analytics for Extreme Scale Climate Science," Supercomputing Conference 2014 (in submission).
Paper	EG Stephan, et al., "A Linked Fusion of Things, Services, and Data to Support a Wind Characterization Data Management Facility," *Journal of Web of Things* (in submission).
Paper	CA Steed, et al., "Big Data Visual Analytics for Exploratory Earth System Simulation Analysis," *Computers & Geosciences*, **61**, 71–82, 2013. http://dx.doi.org/10.1016/j.cageo.2013.07.025 CAGEO 2013 Best Paper Award.
Paper	DN Williams, et al., "Ultrascale Visualization of Climate Data," I*EEE Computer Magazine*, **46**:9, 68–76, September 2013. DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/MC.2013.119.
Paper	L Cinquini, et al., "The Earth System Grid Federation: An Open Infrastructure for Access to Distributed Geospatial Data," I*EEE Future Generation Computer Systems*, September 17, 2013, http://dx.doi.org/10.1016/j.future.2013.07.002.

A listing of additional community outreach activities can be found in **Section 5** and **Appendix D** of the "*Ultrascale Visualization Climate Data Analysis Tools: Three-Year Comprehensive Report*".