UV-CDAT 2013 Near-term Plans and Directions

Overall Plans and Directions

- Reduce executable size and stress test for system performance on LCFs and other platforms
- Review and refine graphical user interface and workflow
- Allow fast implementation and integration of external software via “Plug-in Wrappers”
- Refine the integration of UV-CDAT as a backend service to ESGF and interact directly with ESGF to allow full client-side usage
- Continue packaging and user documentation
- Present live on-line courses on UV-CDAT and generate as video tutorials hosted on YouTube.
- Provide ultra-scale analysis and visualization services to BER and NASA climate research teams
- Software Quality Assurance

Software Process

- Agile Software Development:
  - UV-CDAT Agile software development benefits from Kitware 's suite of tools such as CMake, CTest, and CDash.
- Sophisticated Build System:
  - Builds packages and their dependencies, tracks build order, logs errors and warnings, and supports grouping of packages by a common attribute.
- Provide a control suite to link documentation, dashboards, and issue tracker
- Improve UV-CDAT build process on Supercomputers
  - Automate the release process
  - Provide nightly binary installers for various platforms on dashboards
  - Package specific complier and linker flags
  - Enable package specific tests
  - Provide a general interface for batch tests
  - Provide a general purpose unit testing framework

Access and Analysis

- Improve ESGF interface based on user feedback
  - Save searches and integrate GridFTP support
- Extend ESGF Web server support to reduce ESGF data movement (i.e., remote data reduction and analysis)
  - Django API (mod-wsgi) server
  - Simple analysis and data reduction to start with
- Development of standard analysis and diagnostics
- U-ReAD
  - Easy integration of scientists’ diagnostics
  - Documentation of integrated code
- User Support
  - Workshop and Conference Tutorials
  - Documentation
- Installation support
  - Smart installation to include only the components or packages you want
  - Continue to support the many flavors of Linux and port to Windows
  - Support ESGF and web-based UV-CDAT and remote analysis
- Refactor certain areas of the codebase making them more maintainable and less prone to errors

VisTrails

- Enhance the user interface to make the design more consistent, simple, and modern
- Extend the VisTrails API to reduce the amount of work for UV-CDAT plot and plugin developers
- Expand the Multi-scale Synthesis and Terrestrial Intercomparison Project work by adding higher-level access in the next UV-CDAT release and broadening its use in other UV-CDAT packages
- Refactor certain areas of the codebase making them more maintainable and less prone to errors

DV3D

- Develop new DV3D plots and features
  - As requested by the scientific user community
- Refine the integration with other UVCDAT components
  - ParaView, ViSUS, EDEN, ParCAT, etc.
- Develop new high performance enhancements
  - Address the challenges of very large datasets
- Further develop the distributed analysis components
  - Interactive hyperwall visualization of climate simulation data
- Develop DV3D/UVCDAT-based scientific workbenches
  - Customized for specific research programs
- Give presentations and workshops
  - Demonstrate how these tools can transform the scientific workflow
- Support users
  - Develop documentation and tutorials

ViSUS Plugin

- Integrate plugin more tightly into the UV-CDAT infrastructure
- Expand processing capabilities to include blending of different time scales
- Provide automatic and potentially on-the-fly data conversion utilities
- Expand ViSUS capabilities to block regular meshes

UV-CDAT ParaView Integration and Spatio-Temporal Pipeline

- Tightly-Coupled Integration
  - ParaView is exposed to UV-CDAT via VisTrails workflow. This design enables provenance.
  - Provides custom interface to Climate Scientist. Adding a new representation is easy. The interface shows individual GUI components of each representation.
  - Supports CDMSVariable. Users can drag-and-drop CDMSVariable on a ParaView plot.
- MOC and MHT
  - Parallel implementation of Meridional Overturning Circulation (MOC) and Meridional Heat Transport (MHT) are both implemented in ParaView.
- Spatio-Temporal Pipeline: UV-CDAT Use Case 1
  - The spatio-temporal pipeline is designed for use on datasets with high temporal resolution, in which a visualization product is generated for each timestep. The temporal dimension is parallelized by splitting the processors into several groups, or "time compartments", and having each time compartment processes a timestep.
  - We show performance results from two tests using POP ocean data. The first tests, performed on Jaguar, show an improvement from 18 minutes to 5 minutes. The second set of tests show an improvement from 8 hours to 1 minute using the spatio-temporal pipeline.
- Spatio-Temporal Pipeline: UV-CDAT Use Case 2
  - The spatio-temporal pipeline can also be used to generate statistics from a set of timesteps. For example, a yearly average can be generated from monthly averages.
  - We show performance results from a climate dataset generated by Michael Wehner. Yearly statistics are generated given monthly values.

Spatio-Temporial

- Spatio-Temporal pipeline is currently implemented in UV-CDAT ParaView
  - Gui wizard to generate run scripts
    - Currently only supports Use Case 1
- Current Milestones
  - Paper on spatio-temporal pipeline
    - Goal: LDAV 2013 submission
  - Gui wizard support for Use Case 2

ParCAT

- Tighter UV-CDAT integration
  - Python wrapper
  - Long-term - Merge with ParaView Spatio-Temporal Pipelining
- Additional features as requested by the user community
  - For example, additional parallelizable functions
  - Ensure ParCAT can process ice and ocean model data sets
- Open source ParCAT

VisIt

- Enabling climate science efforts through a programmable parallel infrastructure within VisIt
- Integrating the research and production code developed by VDX into UV-CDAT
- highlight several new features
  - executing optimized climate science driven visualization code
  - enhancing it with custom R and Python code to extend the infrastructure