28 Corporate Drive
Clifton Park, NY 12065
518.371.3971

SCALABLE DATA MANAGEMENT, ANALYSIS AND VISUALIZATION (SDAV) INSTITUTE

Final Scientific/Technical Report

Award: DE-SC0007440

Period of Performance: February 15, 2012 – February 14, 2017

Report Date: March 28, 2017

Awardee: Kitware, Inc.
DUNS: 01-092-6207
Principal Investigator: Berk Geveci, Senior Director of Scientific Computing
(518) 881-4907
berk.geveci@kitware.com

Prepared for: The U.S. Department of Energy, Office of Science
DOE Program Manager: Ceren Susut
(301) 903-0366
Ceren.Susut-Bennett@science.doe.gov

This material is based upon work supported by the U.S. Department of Energy Office of Science under Award Number DE-SC0007440

Abstract

The purpose of the SDAV institute is to provide tools and expertise in scientific data management, analysis, and visualization to DOE's application scientists. Our goal is to actively work with application teams to assist them in achieving breakthrough science, and to provide technical solutions in the data management, analysis, and visualization regimes that are broadly used by the computational science community. Over the last 5 years members of our institute worked directly with application scientists and DOE leadership-class facilities to assist them by applying the best tools and technologies at our disposal. We also enhanced our tools based on input from scientists on their needs. Many of the applications we have been working with are based on connections with scientists established in previous years. However, we contacted additional scientists though our outreach activities, as well as engaging application teams running on leading DOE computing systems.

Our approach is to employ an evolutionary development and deployment process: first considering the application of existing tools, followed by the customization necessary for each particular application, and then the deployment in real frameworks and infrastructures. The institute is organized into three areas, each with area leaders, who keep track of progress, engagement of application scientists, and results. The areas are: (1) Data Management, (2) Data Analysis, and (3) Visualization. Kitware has been involved in the Visualization area. This report covers Kitware's contributions over the last 5 years (February 2012 – February 2017). For details on the work performed by the SDAV institute as a whole, please see the SDAV final report.

Accomplishments

3.3.1.2 ParaView

During the SDAV life-cycle, we had nine ParaView releases before the end of our project. There have been many major improvements in these releases. Some highlights include:

- Support for data parallel frameworks including Dax, Piston and vtkSMP. VTK-m, which unifies these technologies, will be included in ParaView 5.3.
- Development of a charting framework with support for symbol and equation rendering for annotation, and support for vector graphics output.
- Refactoring of the VTK and ParaView libraries to allow for the creation of smaller subsets of both frameworks. This was driven by the need to derive in situ ParaView libraries that fit the problem.
- Introduction of ParaView Catalyst, a ParaView based in situ library. Since its release, Catalyst have been integrated with DOE codes in many scientific domains, including Climate, CFD, Plasma, HEP, and Shock Physics.
- Numerous improvements to ParaView's visual analytics and quantitative analysis capabilities, including support for scatter plot matrices, parallel coordinates and histograms.
- Development of much improved ParaView and Catalyst User's Guides.
- Integration with ADIOS for post hoc IO as well as in transit visualization.
- Integration with DIY2 for the development of distributed algorithms.

We had six VTK major releases over the last 5 years. These releases included numerous improvements and new features in support of DOE needs, including ParaView and VisIt development. Highlights include a major rewrite of VTK's rendering engine to obtain large performance improvements (10x - 100x) on modern graphics cards and software rendering

This material is based upon work supported by the U.S. Department of Energy
Office of Science, under Award Number DE-SC0007440.

implementations, integration of OpenSWR for improved rendering on Intel Xeon Phi processors, integration with VTK-m, and major improvements to the VTK pipeline infrastructure for improved parallel processing.

We have collaborated with and supported various DOE science teams including

- VPIC: Kinetic plasma simulation, Bill Daughton, LANL, PI. Collaborations included post hoc visualization with ParaView and in situ analysis with Catalyst.
- HACC, Cosmology, Salman Habib, ANL, PI. Collaborations included post hoc visualization with ParaView, feature extraction and tracking algorithm development, in situ analysis with CosmoTools.
- Plasma Surface Interaction (PSI) SciDAC Application Partnership, visualizations of helium bubbles and tungsten cavities from LAMMPS simulations using VTK.
- MPAS: Climate simulation, post hoc and in situ analysis and visualization, algorithm development for feature tracking,
- CAM-SE: Climate simulation, post hoc analysis and visualization,
- OSCon - Optimizing SuperConductor Transport Properties through Large-Scale Simulation, SciDAC
- partnership with BES, deploying in ParaView a set of algorithms for extracting, tracking and visualizing vortex dynamics in large-scale time-dependent Ginzburg-Landau (TDGL) superconductor simulation data. We have also integrated IO functionality to load the simulation data into ParaView.

In conclusion, we have improved both the ParaView and VTK visualization tools to support new architecture features at the LCFs and similar machines. We also customized these tools according the specific needs of multiple scientific communities, helping them to achieve greatly improved efficiency, adding functionality they requested, and in several cases embedding our tools into their scientific frameworks. The collaboration with the VTK-m effort has been highly successful. Looking forward, as more algorithms are added to VTK-m to take advantage of multi-core and many many-core architectures, ParaView will automatically benefit from these new capabilities.

Below are specific milestones Kitware contributed to, with specific contributions from Kitware mentioned.

Related Milestones

3.3.1 Year 1: Enhance VisIt and ParaView to leverage multiple cores within a single MPI task

Kitware has demonstrated the use of coarse-grained (parallelize over data blocks) and fine-grained (parallelize over points or cells) parallelism using a multi-threaded approach in VTK and ParaView. We have improved several VTK algorithms to be fine-grained parallel and enabled others to run as coarse-grained parallel algorithms.

3.3.1 Year 2: Develop Integrate VisIt and ParaView with ADIOS for post-processing

Kitware has integrated into ParaView support for the ADIOS IO framework for use with visualization data structures. This has been done primarily by creating a schema specific to VTK

This material is based upon work supported by the U.S. Department of Energy
Office of Science, under Award Number DE-SC0007440.

for mapping its data structures into the ADIOS no-XML API. We can currently read and write unstructured grids, structured grids, and polygonal data types in parallel with multiple time steps. Throughput and performance testing has been primarily through ParaView Catalyst with *in situ* visualization with jobs of 10k to 50k cores.

3.3.1 Year 3: Demonstrate and evaluate *in situ* analysis methods with VisIt and ParaView

Kitware has released ParaView Catalyst, an in situ analysis library based on VTK/ParaView. Its goal is to provide a flexible and modular set of tools that the simulation developers use to instrument their applications with in situ analysis capabilities. We have integrated Catalyst into several DOE simulation codes.

3.3.1 Year 4: Prototype integration of VisIt and ParaView with VTK-m

See 3.3.2 year 5.

3.3.1 Year 5: Integrate VisIt and ParaView with ADIOS for co-visualization

Kitware developed a ParaView plugin to process staging data generated by ADIOS without being stored on disk. This in transit workflow was tested using the CartIso mini-app on Cooley at ALCF running on 72 MPI processes on 6 nodes and ParaView's pvbatch executable running on 24 MPI processes on 2 nodes.

3.3.2.1 VTK-m

When SDAV began in 2012, the scientific visualization community was just beginning to develop algorithms and software that run well on multi-core CPUs or many-core accelerators like GPUs. We identified 3 key projects addressing scientific visualization on multi/many-core processors: Piston, an ASC project lead by LANL, Dax, an ASCR project lead by SNL, and EAVL, an LDRD project lead by ORNL. SDAV adopted these products under its umbrella of tools.

Our evaluation of these tools revealed that each addressed a unique aspect of the problem. Piston focused on efficient and portable algorithms. Dax provided a top-down framework that simplified development. EAVL provided advanced data structures. Our initial work considered the integration of these tools. Although we did manage some forms of integration, the results were suboptimal. The integration was not tight enough to realize the full potential and ultimately we were in danger of repeating each other’s code.

In response, we considered the feasibility of a tighter integration where the code was redesigned under a single software project. The plan required each software team to transfer their software technology to a new, unified project. Such consolidation of software is rare, but fortunately SDAV was structured to include as key personnel the PIs for each of these three projects. This gave us the motivation and organization to form a tight collaboration in which we all agreed to put aside our previous software and move to a new unified software base. Although much of the development of VTK-m was performed under a following ASCR project, this collaboration could never have formed without SDAV, and we consider this an important success of SDAV.

VTK-m version 1.0 was released June 7, 2016. This release includes the integration of the three predecessor toolkits, and with it we demonstrated an efficient combination of algorithm

This material is based upon work supported by the U.S. Department of Energy
Office of Science, under Award Number DE-SC0007440.

implementation, performance portability across devices, advanced data models, and high level scientific visualization building blocks. More information about VTK-m is available on our web site (http://m.vtk.org).

During the run of SDAV we had two VTK-m code sprints: one in September 2015 at LLNL and one in August 2016 at Kitware. Among the two we had many participants that represented work from many different organizations including national laboratories (SNL, LANL, ORNL, LBNL, LLNL), universities (Oregon, UC Davis), and industry (Kitware, NVIDIA, Intelligent Light). These events allowed us to reach out to several interested developers to get them kick started with VTK-m development and also allowed us to make progress in several key areas of VTK-m and its algorithms.

VTK-m development continues to progress and is currently the only viable solution for DOE's scientific visualization needs on multi- and many-core devices, and we expect much of future DOE research to contribute back to the VTK-m library. Furthermore, the DOE Exascale Computing Project (ECP) includes the development of several algorithms in VTK-m to replace the single threaded and MPI-only counterparts in VTK.

Below are specific milestones Kitware contributed to, with specific contributions from Kitware mentioned.

Related Milestones

3.3.2 Y1: Enhancements to existing multi/many-core technologies in anticipation of in situ analysis use cases within LCF codes
3.3.2 Y2: Deployment and evaluation of existing technologies in prototype form

Kitware lead the integration of Piston into VTK and ParaView as well as contributed to Piston development. Kitware contributed to the development of Dax as well as its integration into VTK.

3.3.2 Y3 Initial implementation of VTK-m with an integration of existing packages

Kitware contributed to the development of the initial VTK-m framework. A demonstration of integrating PISTON algorithms with the Dax device layer was made, and a plan for bridging the run-time polymorphism of EAVL data types with compile-time templates in Dax and PISTON was formed. As part of this plan, we have created a prototype set of classes that bridges the run-time polymorphism used by EAVL with the compile-time templates used in Dax and PISTON.

3.3.2 Y4 User community education and migration to VTK-m

During this period, we continued the development of VTK-m. PISTON, Dax and EAVL were fully transitioned to VTK-m and the development continued on a unified front. Developers began to use VTK-m to implement algorithms, and several have reached the point of being contributed to VTK-m proper. Those integrated in this period are cell average, point elevation, Marching Cubes (on regular grids), vertex clustering (for grid decimation), clip, and external faces.

3.3.2 Y5 VisIt and ParaView fully transitioned to use VTK-m in multi/many-core environment

This material is based upon work supported by the U.S. Department of Energy
Office of Science, under Award Number DE-SC0007440.

ParaView 5.3 which was released in March 2017 supports VTK-m integration. All existing VTK-m filters are exposed through the ParaView interface as well as ParaView Catalyst for in situ analysis. Both CUDA and TBB backends are support and therefore GPUs and CPUs are both supported.

This material is based upon work supported by the U.S. Department of Energy
Office of Science, under Award Number DE-SC0007440.