

## LA-UR-16-26987

Approved for public release; distribution is unlimited.

Title: 2016 CSSE L3 Milestone: Deliver In Situ to XTD End Users

Author(s): Patchett, John M.  
Nouanesengsy, Boonthanome  
Fasel, Patricia Kroll  
Ahrens, James Paul

Intended for: Report

Issued: 2016-09-13

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# 2016 CSSE L3 Milestone: Deliver In Situ to XTD End Users

---

John Patchett, Boonthanome Nouanesengsy, Patricia Fasel, James Ahrens

## Summary

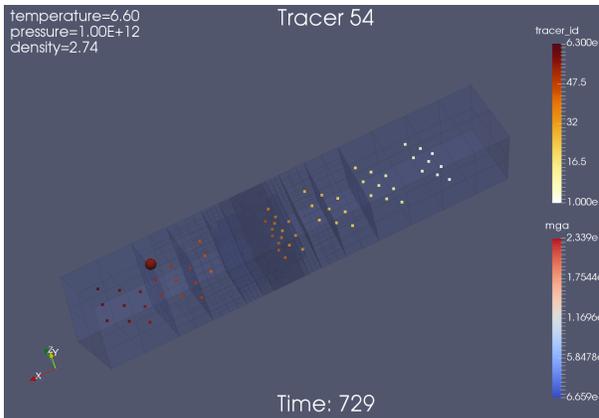
This report summarizes the activities in FY16 toward satisfying the CSSE 2016 L3 milestone to deliver in situ to XTD end users of EAP codes. The Milestone was accomplished with ongoing work to ensure the capability is maintained and developed. Two XTD end users used the in situ capability in RAGE. A production ParaView capability was created in the HPC and Desktop environment. Two new capabilities were added to ParaView in support of an EAP in situ workflow. We also worked with various support groups at the lab to deploy a production ParaView in the LANL environment for both desktop and HPC systems. In addition, for this milestone, we moved two VTK based filters from research objects into the production ParaView code to support a variety of standard visualization pipelines for our EAP codes.

## XTD End Users

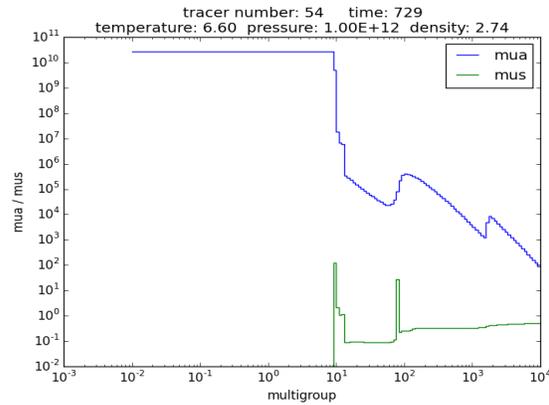
Two XTD end users were selected to help satisfy the requirements of the milestone. These users were Todd Urbatsch and Galen Gisler. Will Dearholt, an XCP end user, also helped. Each of the end users had a specific success with the use of the ParaView in situ capability.

## Use Case I: Debugging

Todd Urbatsch agreed to work with us in order to pursue the ability to study opacities at the locations of Lagrangian tracers during the simulation run. The opacity information is a large variably sized vector, 122 in this case, that makes it difficult to produce full data sets for post processing. We augmented the ParaView Catalyst adaptor to support extracting information for the set of Lagrangian tracers that exist in the simulation. We also added support for extracting the opacity information used in that time step to advance the simulation. Combined, this allowed us to create a set of visualizations that showed both a tracer location and the opacity information tied together for each Lagrangian tracer – a capability that they did not have before.



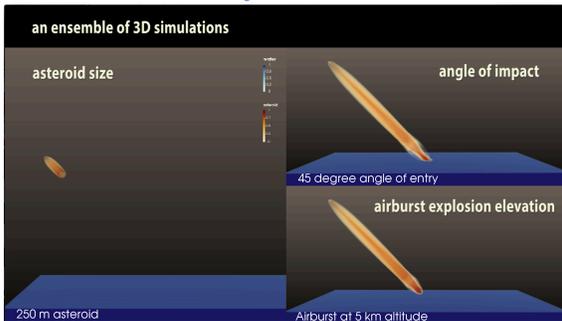
**Figure 1** Plot of selected tracer, displays location, temp, pressure, density and location of other tracers in the simulation space.



**Figure 2** Plot of absorption and scattering at the highlighted Lagrangian point.

We provided Todd Urbatsch with a set of imagery in the form of a Cinema (Ahrens, 2014) database. Figure 1 shows an example of the Urbatsch problem. There are many tracers, each of which is represented as sphere glyphs and colored by ID (1-63). The large red sphere represents the tracer chosen and Figure 2 shows the detailed absorption and scattering information at that tracer location. With this solid foundation capability in place, we continue to work with the user to enable him to create the data products in a flexible manner on his own.

## Use Case II: Analysis of Asteroid – Earth Interactions



**Figure 3** A screenshot of the SC 2016 Visualization Showcase submission.

Galen Gisler of the XTD asteroids team integrated our in situ ParaView Catalyst capability into his workflow to help in analyzing simulations of asteroids entering the earth's atmosphere and impacting deep ocean water. The simulations were run in order to assess the propensity of these asteroids to generate tsunamis, which could put distant populations in peril after such an impact. NASA is interested to know the lower size limit of dangerous asteroids, in order to focus resources on finding and tracking the most dangerous near earth objects. Most of the planet's surface is ocean, so that's where asteroids are most likely to fall. There has been

a serious debate over the last two decades on just how dangerous impact-induced waves or tsunamis are to populated shorelines. A workshop in late August 2016 in Seattle hoped to settle this question, and the simulations done during this L3 milestone contributed to that process. Galen Gisler used visualizations, generated in ParaView with data produced by the ParaView Catalyst capability in xRage, to explain his results and show that simulation results supported his conclusions.

The capability enabled by the in situ adaptor generated much excitement amongst our collaborators and a small team from the University of Texas, with Office of Science funding, a student, with LANL ISTI funding, and a variety of LANL scientists worked together to produce an SC 2016 Visualization Showcase (SC16, 2016) submission based on Galen Gisler's research and simulation of these asteroid impacts in deep ocean water.

## Use Case III: Applied Instruction Testing

Will Dearholt followed instructions on xcp-confluence to exercise the standard workflow that results in in situ imagery output at runtime:

1. Run the simulation with in situ configuration pointing at standard visualization definition file that produces vtk multiblock files.
2. Post Process the generated multiblock files using ParaView to produce a visualization definition file that produces desired imagery when executed in situ.
3. Edit the simulation configuration to use the visualization definition file produced in step 2 and execute the simulation to produce imagery.

Will was able to execute the workflow successfully on one of his production runs using the supplied directions, this showed that instructions maintained on xcp-confluence were adequate for an experienced and motivated user to successfully execute an in situ workflow with the EAP codes. In addition this process exposed an absolute need for the X division network and support infrastructure to support the use of ParaView as an approved application.

## Production ParaView

A ParaView client running on the user's desktop is a necessary part of the in situ workflow. The ParaView client version must match the parallel ParaView server version running on the supercomputer. A problem was identified in both the yellow and red networks: not all end users were able to easily get a proper ParaView client version on their desktop machines. The result was that every user needed to make a custom request to their desktop support staff for a specific version of ParaView to be installed. We worked with various groups and individuals at the Laboratory and ParaView packagers at Kitware to support the institutional desktop computing systems ability to roll out multiple and simultaneously supported versions of ParaView. This required talking with different groups of LANL Information Technology staff, gathering requirements, and feeding those requirements back to the packaging efforts at Kitware. We were able to walk through this effort and get a nearly complete deployment of ParaView 4.3.1 and are currently working on ParaView 5.1.2 release.

The effort of coordination involved in delivering ParaView to HPC systems and Desktops has resulted in establishment of a more rigorous release schedule for ParaView, as well as regular meetings and long term planning amongst LANL, Sandia, and Kitware to ensure that capability and the rollout of future ParaView versions will be increasingly smooth.

## Production ParaView on Supercomputers

All LANL supercomputers provide modules for ParaView built against a variety of compilers and MPI versions that are deemed important to the EAP development team at the time of the release. A "superbuild" of ParaView is used for this purpose. The superbuild contains an automated mechanism for building a variety of ParaView dependencies to more easily integrate the parallel ParaViews into our production-computing environment. The code for producing these superbuilds is stored in git repositories at <https://gitlab.kitware.com/paraview/paraview-superbuild>.

## Interaction between Desktop and Supercomputer

The HPC Consultant and Parallel Tools teams developed an automated connection system between the desktop and the supercomputer. The setup and configuration of this automated system can be found at [https://hpc.lanl.gov/paraview\\_usage](https://hpc.lanl.gov/paraview_usage), these instructions substantially ease the burden on a user who has data residing on a LANL supercomputer and wants to perform visualization or analysis activities using ParaView.

## Desktop ParaView

LANL prefers specific packaging for software distributions that allow their automated systems to automatically install, update, validate, and remove specific software packages. There are many details that we have or are dealing with including proper signing of mac binaries using a certified apple developer ID and produce pkgs with versioning systems that allow multiple versions rather than upgrades through LANL's Casper system. For windows clients we are working to solve a

problem where LANL’s automated package testing uses virtual machines with insufficient OpenGL library versions. Having to match the software versions across HPC and desktop environments is a new problem for us and we are committed to ensuring that LANL Supported ParaView versions are accessible to our end users.

## Delivery of Research Projects to Production

The ideas, mathematical models, and proofs of concepts generated in the CSSE R&D efforts are frequently found to have direct and immediate applicability and usefulness to the ASC program production environments. For these research products to move from research and into a production capability, typically, development and integration needs to occur. For the purposes of this milestone we identified two basic sets of functionality that we believed would benefit the general usefulness of the ParaView Catalyst in situ capability. The first is a ghost cell generator filter for arbitrary well-formed unstructured grids that alleviates the absolute need to generate ghost cells in the in situ adaptor. The second is the ability to resample a costly unstructured grid representation as structured data. Both of these capabilities are now in the current ParaView as of ParaView 5.1.2.

### Ghost Cell Generator Filter

**Problem:** ParaView Catalyst, which is often used for in situ processing, requires ghost cells, replicated cells on parallel decomposition boundaries, to be calculated by the parallel code itself. Frequently, codes like Rage only have native ghost cells for polyhedral cell faces, not cell edges and points. Ghost Cells for faces, edges and points are necessary for many visualization filters like surface and contour to produce accurate results at the parallel decomposition boundary.

**Solution:** We developed a parallel algorithm to calculate two layers of ghost cells on generalized well-formed unstructured grids. We have delivered this capability in the Visualization Toolkit (VTK), the basis for ParaView, Catalyst and VisIT among other applications as vtkPUnstructuredGridGhostCellsGenerator. This capability will soon be exposed in these higher level applications. This capability was developed specifically for Rage, but we expect to use it, unchanged, for other LANL codes in the future.

**Result:** Domain scientists using VTK based visualization tools will, simply call the ghost cell generator prior to any parallel filters that require ghost cells. Simulation code developers will not need to produce ghost cells in their in situ adaptors, making adaptor development faster and easier. Importantly, domain scientists can depend on a complete set of ghost cells. Timings for the filter can be seen in Figure 4.

| <b>Time In Seconds to Calculate Two Full Ghost Cell Layers</b> |  |                      |                        |
|--|--|----------------------|------------------------|
|  | <b>Scaling factor of asteroid impact input</b> |                      |                        |
| <b># processors</b>  | <b>0.5 (2MCells)</b>                           | <b>1.0 (4MCells)</b> | <b>1.5 (6.5MCells)</b> |
| 16   | 1.181  | 1.960                | 2.740                  |
| 32   | 1.110  | 1.907                | 2.255                  |
| 64   | .973   | 1.612                | 2.453                  |

Figure 4 Performance of the ghost cell generator filter in ParaView.

### Resample to Image Data

Sampling large unstructured grids to produce structured data enables a variety of visualization operation like faster volume rendering and a compact representation of the data allowing users to work with a data storage budget while managing resolution of results. A Kitware blog post about these capabilities can be read at <https://blog.kitware.com/dataset-resampling-filters/>. Resample to image data filter can have its data stored on disk, but its output is a structured data set balanced across all of the processors. This could then be saved to disk or used to generate data products such as a volume rendering.

## Conclusion

The FY 2016 CSSE L3 Milestone to deliver in situ to XTD end users of EAP codes has been successful. It has also exposed an ongoing need to continue working with XTD end users, XCP and CCS developers in addition to LANL infrastructure support in HPC and other groups. The Milestone is ending with this understanding and commitment among the various groups to maintain and improve upon the production capability of ParaView and ParaView catalyst. The end users who participated are generally happy and willing to continue working with us into the future to improve upon the current state.

## Acknowledgements

Many people contributed and supported the work presented here. We would like to acknowledge the following for their efforts: Todd Urbatsch, Galen Gisler, William Dearholt, David Rogers, Jennifer Green, Giovanni Cone, Ekkehard Koch, Drea Valdez, Charles Atkins, Linn Collins, Sriram Swaminarayan, and Joann Campbell.

## Bibliography

Ahrens, J. (2014). An image-based approach to extreme scale in situ visualization and analysis. *SC '14 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 424-434). Piscataway: IEEE Press . SC16. (2016, 09 07). *SC16 Scientific Visualization Showcase for Submitters*. Retrieved 09 07, 2016, from Supercomputing 2016: <http://sc16.supercomputing.org/submitters/showcases/scientific-visualization-showcase/>