

2.1.2 SNL ATDM Visualization

Overview The SNL ATDM Data and Visualization work consolidates existing ATDM activities in scalable data management and visualization. Part of the responsibilities of the SNL ATDM Data and Visualization Project is the maintenance and development of visualization resources for ATDM applications on Exascale platforms. The ATDM Scalable Visualization project provides visualization and analysis required to satisfy the needs of the ASC/ATDM applications on next-generation, many-core platforms. This involves many activities including the re-engineering of visualization algorithms, services, and tools that enable ASC customers to carry out data analysis on ASC systems and ACES platforms. Current tools include scalable data analysis software released open source through ParaView [7], VTK [8], and Catalyst [9]. We are also both leveraging and contributing to VTK-m [10], a many-core visualization library, to satisfy our visualization needs on advanced architectures.

The scope of the Scalable Visualization under ATDM at SNL is R&D for the programming model and implementation of visualization code for ASC/ATDM projects and ASC/ATDM application support.

Key Challenges The scientific visualization research community has been building scalable HPC algorithms for over 15 years, and today there are multiple production tools that provide excellent scalability [7, 9]. That said, there are technology gaps in data analysis and visualization facing ATDM applications as they move to exascale. As we approach exascale, we find that we can rely less on disk storage systems as a holding area for all data between production (by the simulation) and consumption (by the visualization and analysis). To circumvent this limitation, we must integrate our simulation and visualization into the same workflow and provide tools that allows us to run effectively and capture critical information.

Solution Strategy The SNL ATDM Visualization effort is addressing its challenges by leveraging and expanding the following technologies.

1. **In situ visualization** We are integrating our HPC visualization capabilities into our ATDM simulations by leveraging the Catalyst *in situ* visualization library [9]. Catalyst is part of the code supported by the ALPINE project (WBS 2.3.4.12), and we leverage this software by integrating it with ATDM application codes and applying it in our simulation runs.
2. **Data compression** Data compression allows us to make more effective use of the bandwidth and capacity of our storage systems. We are developing the TuckerMPI library [11] to provide compression specifically optimized for distributed parallel simulation data.
3. **Algorithm portability** Running visualization *in situ* with a simulation mandates executing the visualization algorithms on the same compute resources. Thus, our visualization software must be able to run on ASC systems and ACES platforms. We are leveraging the VTK-m software [10] being developed by the ECP/VTK-m project (WBS 2.3.4.13), optimizing the library for ACES platforms, and integrating the code with our ATDM software.

Recent Progress SNL ATDM has several ongoing visualization-centric activities.

In Situ Visualization Under ATDM, several Catalyst adapters are under development for Sandia simulation codes. These adapters leverage a range of *in situ* technologies such as shallow copy of simulation data structures for reduced memory footprint, data compression of simulation grid data fields, and the use of VTK-m filters for on-node compute performance.

SPARC (Sandia Parallel Aero sciences Research Code) [12] is a massively parallel compressible CFD code that is designed to be highly scalable. SPARC uses a hybrid structured/unstructured mesh format, and there is ongoing development to construct a Catalyst adapter that efficiently uses memory to represent this data structure using shallow copy data structures provided by ParaView.

Nalu [13] is a massively parallel unstructured low Mach flow code. An adapter for Catalyst was developed for Nalu using the existing Seacas IOSS adapter for the Sierra framework to output unstructured mesh data to Catalyst. Examples of *in situ* visualization with Nalu are shown in Figure 2.

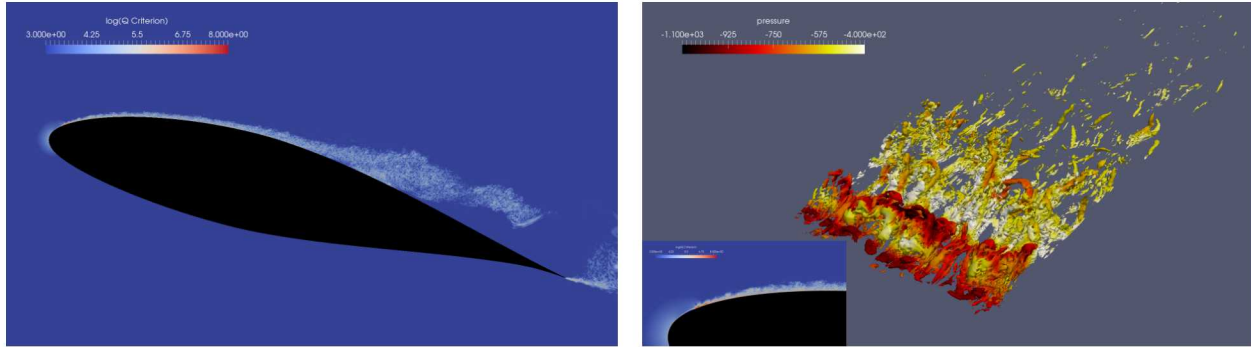


Figure 2: Examples of an *in situ* visualization of a Nalu simulation on 2560 processes of airflow over a wind turbine airfoil. At left: a cross-sectional slice through the airfoil along the direction of flow colored by Q criterion. At right: detail at the leading edge of the wind turbine airfoil.

Sparta [14] is a Sandia developed massively parallel DSMC (Direct Simulation Monte Carlo) code. Sparta uses a unique hierarchical Cartesian grid structure to track simulation particle movement and interaction. The Catalyst adapter implementation for Sparta uses shallow copy data structures to represent this hierarchy as an unstructured grid without forcing unnecessary copies of Sparta data structures.

Data Compression Although the TuckerMPI compression library is already available as an open standalone git repository [15], we pursued three independent approaches to integrate TuckerMPI compression with the *in situ* technologies being developed in parallel: (1) modify Catalyst adapter, integrated with a miniFE demo example, to perform TuckerMPI compression (copy input data, create tensor data structures, perform compression over distributed data), (2) create a standalone “TuckerWriter” plugin for ParaView, packaged as a standalone shared object library (links appropriately with the separately built Tucker Library) that can be loaded from a ParaView GUI, and, (3) extend the SPARC Catalyst direct output branch to work alongside SPARC writer output. This extension allows Catalyst output from SPARC input decks that request output in a specific file format (CGNS), and will be used to construct test cases for TuckerMPI compression.

In addition to TuckerMPI integration with ParaView/Catalyst, we are also conducting R&D to extend the capability to compress unstructured non-rectilinear mesh data using functional tensor approximations. We are researching an approach whereby multi-variate data on an unstructured mesh is treated as a high-dimensional function (spatial dimensions, time), which may be approximated as a small sum of product of univariate functions. Encoding the univariate functions requires storing only a small number of function coefficients/parameters, resulting in large compression.

Algorithm Portability The SNL ATDM project is helping VTK-m run better on ASC platforms with multiple activities. First, we are improving the parallel radix sorting operation in VTK-m. Parallel sorting is a core component of many of our visualization algorithms. Sorting helps us group common identifiers and conditions index sequences for fast searching. Improving the performance of sort within the VTK-m framework implicitly improves the performance of many algorithms that already use it. We are implementing a parallel radix sorting algorithm based on the work of Satish, et al. [16]. Our initial experiments indicate that the radix sort generally outperforms the default TBB sort algorithm typical data types.

Second, we are working to improve VTK-m’s support vector instructions to improve performance on processors such as the Intel Xeon and Xeon Phi. Compilers work to identify code loops where vectorization can be applied, but in VTK-m changes were required to help the compiler. To enable vectorization across multiple complex data types, we are experimenting with loading values in blocks as we schedule an algorithm. This allows the compiler to better vectorize worklet functions that are now accessing well behaved blocks of memory. Given a sufficiently complex worklet, the benefit of better vectorization outweighs the overhead of maintaining blocks of values.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

- [1] B. Smith, R. Bartlett, and xSDK developers. xSDK community package policies, 2016. version 0.3, December 2, 2016, <https://dx.doi.org/10.6084/m9.figshare.4495136>.
- [2] R. Bartlett, J. Sarich, B. Smith, T. Gamblin, and xSDK developers. xSDK community installation policies: GNU Autoconf and CMake options, 2016. version 0.1, December 19, 2016, <https://dx.doi.org/10.6084/m9.figshare.4495133>.
- [3] Todd Gamblin, Matthew P. LeGendre, Michael R. Collette, Gregory L. Lee, Adam Moody, Bronis R. de Supinski, and W. Scott Futral. The Spack Package Manager: Bringing order to HPC software chaos. In *Supercomputing 2015 (SC'15)*, Austin, Texas, November 15-20 2015. LLNL-CONF-669890.
- [4] Alquimia Project Team. Alquimia Web page. <https://bitbucket.org/berkeleylab/alquimia>.
- [5] PFLOTRAN Project Team. PFLOTRAN Web page. <http://www.pflotran.org>.
- [6] Alicia Marie Klinvex. xSDKTrilinos user manual. Technical Report SAND2016-3396 O, Sandia, 2016.
- [7] James Ahrens, Berk Geveci, and Charles Law. ParaView: An end-user tool for large data visualization. In *Visualization Handbook*. Elsevier, 2005. ISBN 978-0123875822.
- [8] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit: An Object Oriented Approach to 3D Graphics*. Kitware Inc., fourth edition, 2004. ISBN 1-930934-19-X.
- [9] Utkarsh Ayachit, Andrew Bauer, Berk Geveci, Patrick O'Leary, Kenneth Moreland, Nathan Fabian, and Jeffrey Mauldin. Paraview catalyst: Enabling in situ data analysis and visualization. In *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV 2015)*, pages 25–29, November 2015.
- [10] Kenneth Moreland, Christopher Sewell, William Usher, Li ta Lo, Jeremy Meredith, David Pugmire, James Kress, Hendrik Schroots, Kwan-Liu Ma, Hank Childs, Matthew Larsen, Chun-Ming Chen, Robert Maynard, and Berk Geveci. Vtk-m: Accelerating the visualization toolkit for massively threaded architectures. *IEEE Computer Graphics and Applications*, 36(3):48–58, May/June 2016. DOI 10.1109/MCG.2016.48.
- [11] Woody Austin, Grey Ballard, and Tamara G. Kolda. Parallel tensor compression for large-scale scientific data. In *IPDPS'16: Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium*, pages 912–922, May 2016.
- [12] Micah Howard, Andrew Bradley, Steven W. Bova, James Overfelt, Ross Wagnild, Derek Dinzl, Mark Hoemmen, and Alicia Klinvex. Towards performance portability in a compressible cfd code. In *AIAA Computational Fluid Dynamics Conference*, AIAA AVIATION Forum, 2017.
- [13] Nalucfd webpage. <http://nalucfd.org/>.
- [14] M. A. Gallis, J. R. Torczynski, S. J. Plimpton, D. J. Rader, and T. Koehler. Direct simulation monte carlo: The quest for speed. In *29th Intl Symposium on Rarefied Gas Dynamics*, AIP Conference Proceedings, 2014. <http://sparta.sandia.gov>.
- [15] TuckerMPI. <https://gitlab.com/tensors/TuckerMPI>.
- [16] Nadathur Satish, Changkyu Kim, Jatin Chhugani, Anthony D. Nguyen, Victor W. Lee, Daehyun Kim, and Pradeep Dubey. Fast sort on cpus and gpus: A case for bandwidth oblivious simd sort. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10)*, pages 351–362, 2010.