Final Report

Scalable Analysis Methods and In Situ Infrastructure for Extreme Scale Knowledge Discovery

Prepared for: U.S Department of Energy, Office of Science, Advanced Scientific Computing

Research Grant Number: DE-SC0012387

Prepared by: Kitware, Inc.

DUNS: 01-092-6207

Principal Investigator: Dr. Patrick O'Leary
Assistant Director of Scientific Computing
(518) 371-3971 x621
patrick.oleary@kitware.com

Submission Date: September 13, 2017

Submitting Official: Brenda McCormack
Reporting Administrator
(518) 371-3971 x128
brenda.mccormack@kitware.com

Abstract/Executive Summary

The primary challenge motivating this project is the widening gap between the ability to compute information and to store it for subsequent analysis. This gap adversely impacts science code teams, who can perform analysis only on a small fraction of the data they calculate, resulting in the substantial likelihood of lost or missed science, when results are computed but not analyzed. Our approach is to perform as much analysis or visualization processing on data while it is still resident in memory, which is known as in situ processing. The idea in situ processing was not new at the time of the start of this effort in 2014, but efforts in that space were largely ad hoc, and there was no concerted effort within the research community that aimed to foster production-quality software tools suitable for use by Department of Energy (DOE) science projects. Our objective was to produce and enable the use of production-quality in situ methods and infrastructure, at scale, on DOE high-performance computing (HPC) facilities, though we expected to have an impact beyond DOE due to the widespread nature of the challenges, which affect virtually all large-scale computational science efforts. To achieve this objective, we engaged in software technology research and development (R&D), in close partnerships with DOE science code teams, to produce software technologies that were shown to run efficiently at scale on DOE HPC platforms.

The main focal points of our activities were as follows:

- Show the viability of in situ methods and infrastructures, at scale, on contemporary DOE HPC platforms within the context of several DOE science exemplars. Our approach consisted of annual campaigns targeting runs, at increasing scale, on multiple DOE HPC platforms. We used a combination of mini-applications and science codes and focused on a limited set of design and execution patterns to limit problem scope.
- To increase code portability through the development and use of a generic interface between data producers (science codes) and data consumers (in situ methods and infrastructures).
- Engage in close partnerships with DOE science code teams, who provided scientific-based motivation for the software technology R&D and whom also benefited through the integration of this software to utilize associated in situ methods and infrastructures.
- Through outreach, publications, and presentations, foster the coordination and formation of a community of in situ processing developers, researchers, and practitioners.
- Produce and release software artifacts, consisting of the generic in situ interface, methods, and infrastructures.

Figure 1 presents an overview of the team's accomplishments shown in timeline form. This chart shows yearly campaigns where we push the limits of scalability, first to 64K ranks in early 2015, to 256K ranks in 2016, and followed by 1M ranks in 2016.

Over the course of the project, we have been highly productive regarding generating publications:

- Seven peer-reviewed journal articles (IEEE Visualization; Computer Graphics Forum; Parallel Computing; Computational Astrophysics and Cosmology);

- Nineteen peer-reviewed conference papers (SCXX, HPDC, IEEE Cluster, Grid, and Cloud Computing; eScience; Parallel CFD; ALENEX; IPDPS; and many others);
- Eight peer-reviewed workshop papers; and
- Four peer-reviewed panel abstracts (LDAV, SCXX, SIAM PP16, SIAM CSE 2017).

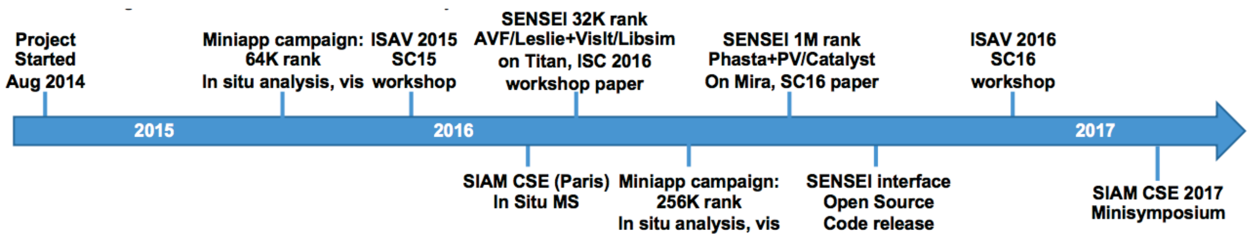

Figure 1: Timeline of Extreme-scale *in situ* methods and infrastructure (SENSEI) team accomplishments, beginning with the project start in August 2014, and going through May 2017.

The team has also been highly active in outreach and fostering interaction within the community of in situ researchers and practitioners outside this project. We launched a new workshop at the SC annual conference series, In Situ Infrastructures for Enabling Extreme Scale Analysis and Visualization (ISAV), held in 2015 and 2016, and currently accepted for the 2017 program. This workshop has been an overwhelming success, drawing the attention and participation of the worldwide HPC in situ community, with the involvement from academia, government labs, and industry. We have also organized multiple mini-symposia at SIAM meetings, presented tutorials at SC in 2015, 2016 and 2017, engaged in direct outreach to undergraduates and underrepresented groups, and given numerous invited presentations on various topics related to in situ technologies.

Introduction

Problem

For software tools to be usable by DOE scientists and engineers on DOE HPC platforms, they must undergo a process of scaling verification on those platforms. While each of VisIt/Libsim, ParaView/Catalyst, Adaptable IO System (ADIOS), and GLEAN were, to varying degrees, production-quality tools, but they had not undergone a verification process rigorously on DOE HPC platforms. Upon successful completion of such tests, the DOE science and engineering community — including science code teams as well as DOE HPC facilities personnel — will have greater confidence in the software infrastructure being deployed and used on their systems. Such confidence helps to promote more widespread adoption and use within the DOE science community.

Approach

In brief, our approach to this problem was to engage in systematic scalability studies on all the key DOE HPC facilities. These studies consisted of using a mini-application as a data producer connected to in situ data consumers and conducting runs at increasing scale over the lifetime of the project.

We opted for an approach like that adopted by many DOE HPC facilities for both system requests for information (RFIs) as well as system acceptance tests: the use of mini-applications that are representative of typical scientific workloads. In our case, the problem space is a bit more complicated because there is a data producer as well as a data consumer.
Running mini-application campaigns at gradually increasing scale over the three-year life of the project, across several DOE platforms, allowed us to discover impediments to scalability, as well as to design and engineer solutions to those obstacles. Those code improvements, in turn, found their way into production releases of the constituent in situ technologies, and in turn, benefit a worldwide scientific community.

Accomplishments

In the last three years, we conducted two major scalability studies designed to push the limits of concurrency and performance on DOE HPC platforms. To prepare for these studies, we created and assembled the necessary software components, which consist of data producers, the in situ interface, and data consumers.

For a data consumer, we created a standalone message passing interface (MPI) based code, oscillators, that computes a time-varying function on a three-dimensional (3D) computational domain that models the behavior of a user-specified number of damped oscillators. This code uses a 'shared nothing' form of parallelism, where the MPI ranks can each perform computations independently where no inter-rank communication is required. As such, it is highly scalable to arbitrary concurrency. Over the lifetime of the project, we instrumented this mini-application to use the SENSEI generic in situ interface. As with other types of mini-applications, our mini-application is representative of many types of DOE science and engineering workloads, namely a bulk-synchronous parallel (BSP) computation built atop of MPI. The oscillators code allows us to replicate BSP computation at scale with a much lower computational cost compared to a production science code.

We created several in situ data consumers or endpoints. Two fairly straightforward end-points are a constant-width-bin histogram computation and a time-varying autocorrelation. While single time step in situ calculations are important, including a time-varying in situ calculation (a first) is one of relevance to many DOE science and engineering problems. Other in situ data consumers include visualization pipelines, implemented in both VisIt/Libsim and ParaView/Catalyst, that create images of a heatmap-colored slice extracted from a 3D volume, and rendered from a 3D viewpoint.

All of the software tools that we have built for use in the mini-application campaigns are released under an open source license and are available to the worldwide scientific community (https://gitlab.kitware.com/sensei/sensei).

Spring-Summer 2015 Scalability Study

This campaign consisted of weak-scaling studies at concurrency levels ranging from 1K to 64K-way concurrency on Mira at ALCF, Edison at NERSC, and Titan at OLCF. In these tests, we performed in situ and in transit analysis of the histogram and autocorrelation calculations. VisIt/Libsim and ParaView/Catalyst performed the in situ calculations (no data movement), while ADIOS and GLEAN performed in transit calculations (data moves from the computational cores to the analysis cores, but never lands on disk). These studies revealed several impediments to calculations at scale, such as potentially long initialization times due to the use of shared libraries. Also, we encountered problems with ADIOS on the Blue Gene platform and ultimately decided to not invest resources in the engineering required for ADIOS to work on the Blue Gene platform because that platform was known to not be on the list of next-generation systems at DOE HPC centers.

Spring-Summer 2016 Scalability Study

As with the Spring-Summer 2015 scaling study, we made use of the oscillators code as a data generator, but this time leveraged the new SENSEI generic in situ interface. This approach allowed us to connect the oscillators code to any of the infrastructures without modification to change from one to another. This weak scaling study pushed the limits out to 256K-way concurrency on Titan and Mira (NERSC did not have a machine that large at that time). This campaign went into significant depth to measure the impact of the SENSEI generic interface regarding runtime and memory footprint at scale, as well as compared the cost of running a representative at-scale workload in post hoc fashion to in situ fashion.

Science Codes At-Scale

We engaged in at-scale runs of several different science codes with extremely high levels of concurrency. One such science code was the parallel, hierarchic (second to fifth order accurate), adaptive, stabilized (finite-element) transient analysis tool for the solution of compressible or incompressible flows (PHASTA https://fastmath-scidac.llnl.gov/software/phasta.html) code.

During one particular test, we connected PHASTA to ParaView/Catalyst at over 1M-way concurrency, which was an unprecedented standard of scalability for in situ processing.
The results from the Spring-Summer 2016 and Science Codes At-Scale scalability studies provided data for a technical paper that appeared in the 2016 International Conference for High-Performance Computing, Networking, Storage and Analysis (SC16) Technical Program.

Results

SENSEI Generic In Situ Interface

Problem

The key issue for DOE science codes is that the code instrumentation needed to take advantage of in situ methods and infrastructures limit a code to one specific approach and implementation. If a science code desires to utilize in situ processing capabilities, the code developers must modify the software to make calls to a particular in situ infrastructure. Over the course of the code's lifetime, if the developers want to try out a different in situ method or infrastructure, then the code must be modified to make use of the new in situ infrastructure's application programming interface (API). This example demonstrates the cost and complexity of instrumenting and maintaining science codes.

A related problem is the fact that there exists a diversity of in situ methods and infrastructures. On this project alone, we have representatives from four different existing in situ infrastructure teams (VisIt/Libsim, ParaView/Catalyst, ADIOS, GLEAN) as well as researchers who develop numerous in situ analysis methods for everything from statistics to topology. The presence of so many different ways of making use of in situ methods and infrastructures is itself an impediment to their use, as well as a barrier to innovation and broad use in the DOE science community. This difficulty again illustrates the cost of developing and maintaining science codes.

Approach

One of the central objectives of this project was to design and create a generic in situ interface, which enabled the instrumentation of a science code once, but connect to and make use of any number of potential in situ methods and infrastructures (aka - *write once and use everywhere*). A central challenge with this objective is mapping a science code's data model to the in situ method's or infrastructure's data model.

Our approach to the write once and use everywhere problem is to provide means for mapping from a science code's data model, via data adaptor, into an in situ data bridge, and thence into the in situ method's or infrastructure's data model via an analysis adaptor. This approach is adaptable to data models common in DOE science codes and a diverse set of in situ processing methods and infrastructures.

Accomplishments

We implemented the SENSEI generic in situ interface (see Figure 2). The first element is the data adaptor between the data producer and the in situ bridge, where we map the data producer's data model into a generic data object. The second element is the analysis adaptor between the in situ bridge and the data consumer, where we map the contents of the generic data object into the data consumer's data model.

We instrumented the oscillators mini-application to utilize the SENSEI interface for the Spring-Summer 2016 scalability studies, and demonstrated running a data producer and connecting it to several different in situ methods and infrastructures. This accomplishment demonstrated us reaching our objective of having a data producer being able to make use of different in situ methods and infrastructures through a single interface.

The SENSEI generic in situ interface was the subject of a peer-reviewed publication at the In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV http://vis.lbl.gov/Events/ISAV-2016/) workshop held in conjunction with the International Conference on High-Performance Computing, Networking, Storage, and Analysis (SC16) in Salt Lake City during November 2016.

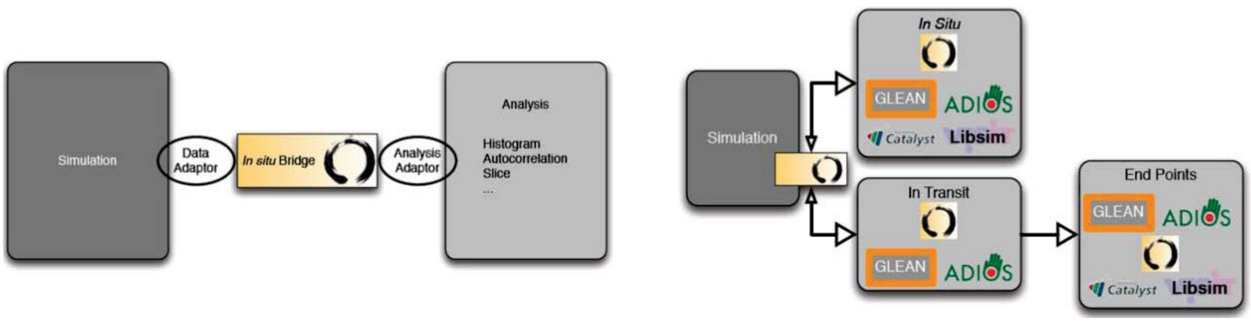

Figure 2: The SENSEI generic in situ interface. (Left) A data adaptor provides the means to map from a science code's data model into a generic data object, which goes through a bridge, where an analysis adaptor maps it into the native data model of an in situ analysis method. (Right) This approach allows us to instrument a science code once with the SENSEI generic in situ interface, then utilize several different in situ methods and infrastructures.

Next Steps

While this approach and implementation proved useful, there remains work to be done. The initial SENSEI generic in situ interface implementation supports mapping two dimensional (2D) and 3D structured arrays into and out of the generic data object, which covers a significant number of DOE science and engineering use cases. However, there are many other data models in DOE science codes, such as adaptive mesh refinement, unstructured meshes, and so forth. Providing support for these more complex data models to be mapped requires additional engineering to encode the complex metadata describing those meshes/data models into the generic data object while making it available to downstream data consumers. This topic is the subject of future work.

Science Code Team Collaborations

Just as predicted in the proposal, SENSEI has been incorporated into other projects in a large number of disciplines. In the sections below, we have highlighted projects enabled through Kitware collaborations.

Aircraft Design

Science Objective and Stakeholders

The design of aircraft wings is a complex process, especially when considering active flow control features such as injecting fluid into the boundary layer. Such flow control features can positively affect airfoil performance by re-attaching separated flow, reducing vibrations and virtual reshaping of the lifting surfaces. The first step in the design process is an understanding of these desired characteristics by scientists and engineers. The next step is trying to optimize airfoil performance through flow control injection placements and wing geometry. This optimization can result in several runs which each requires a significant computational capacity to capture the complex physics properly.

To gain insight promptly, in situ methods can be used to reduce the cycle time in examining different designs for favorable performance. Since the design metrics are known a priori, we can prorate initial costs for setting up the in situ analysis and visualization over the entire ensemble of runs.

PHASTA (https://github.com/PHASTA/phasta) is a general purpose CFD code developed utilized by Ken Jansen and Michel Rasquin of the University of Colorado (UC) at Boulder to examine the given workflow enhanced by HPC computing for Boeing. They demonstrated that PHASTA scaled well to up to 3 million MPI ranks on the Argonne Leadership Computing Facility's' (ALCF) Mira BG/Q supercomputer. This exploration was performed using 4 MPI ranks per core or 64 MPI ranks per node. Each node has 16 GB of memory which results in an average of 256 MB of memory available to each PHASTA MPI process. Jansen and Rasquin focused their work on simulation of a realistic tail rudder assembly.

Our science stakeholders on this are Ken Jansen from UC Boulder, who is also part of the FASTMath SciDAC Institute, and Michel Rasquin from Cenaero in Belgium and UC Boulder.

Figure 3: Zoomed slice through the wing of a PHASTA simulation.

Accomplishments

One of the major goals of this project is to develop production-level capabilities for in situ analysis and visualization at the leading edge of computational capacity. This goal means that our framework needs to run at the scales of the leading edge science codes. PHASTA is a perfect science code match for this goal since it performs well at the scales that we desired to test.

Memory consumption is an important consideration for PHASTA when running on machines such as ALCF's Mira since there is only 256 MB available per MPI rank. Using PHASTA on Mira will require our framework to both scale and use memory efficiently.

PHASTA was one of the earliest codes to be instrumented with the Catalyst in situ infrastructure. Because of this, it was relatively straightforward to modify PHASTA to use the SENSEI generic in situ interface API.

We wrote the original data adaptor before we developed any of the zero-copy infrastructure in VTK. While the memory for PHASTA's scalar arrays could be used directly in VTK's data arrays for the initial integration of the data adaptor, the velocity and point coordinate vector arrays had to be deep copied due to PHASTA using a structure-of-arrays (SOA) memory layout. With the zero-copy functionality within VTK, we modified the data adaptor to reduce the in situ memory consumption. To run efficiently on Mira, we also built an edition of Catalyst enabling the slice analysis adaptor to reduce library size and thus memory footprint. The slice analysis adaptor computes a slice through the domain and then renders an image of the generated slice. The resulting static executable size was 153 MB while the executable size without Catalyst was 87 MB. Finally, we have reduced the file system introspections during Catalyst initialization by performing the file IO on the first MPI process and broadcasting that information to the other processes.

Future Work

Since benchmarking PHASTA runs on ALCF Mira for 256K and 1M MPI rank[1] runs, we have been analyzing the results and planning our collaborative future work.

In order to run PHASTA on ALCF Mira as efficiently as possible, we need to reduce the Catalyst memory footprint such that everything can fit within the 256 MB per MPI rank. This reduction will allow us to scale to even larger process counts to help understand the SENSEI generic in situ interface behavior at full machine scale on ALCF Mira.

Ongoing discussions with Ken Jansen have focused the work on a computational monitoring and steering use-case within SENSEI. We hope to achieve a stretch goal of running full machine scale on ALCF Mira later in 2017.

Climate Science

Science Objective and Stakeholders

Accelerated Climate Modeling for Energy (ACME) project aims to accelerate the development and application of state-of-the-art Earth system science codes. The project will explore and answer science questions in climate change, particularly, biogeochemistry, cryosphere[2], and the

[1] The run exceeding 1M-way concurrency is the largest-ever in situ visualization and was published in our SC16 technical program paper.

[2] Areas of the earth where water exists as ice or snow.

water cycle. ACME achieves its scientific program only through the use of DOE's Leadership Computing Facilities.

ACME science codes like the Model Prediction Across Scales project (MPAS-Ocean) and Community Atmosphere Model (CAM) 5.0 evolve tens to hundreds of parameters and are capable of producing thousands of derived quantities. However, the widening gap between FLOPs and I/O capacity makes full-resolution post hoc analysis expensive, if not impossible. We believe that by integrating ACME science codes with in situ I/O mitigation capabilities, we will enable the ACME project to complete their science objectives.

The overall science stakeholder for this project are ACME and the Community Earth System Model (CESM) science code development teams, in particular, Sandia National Laboratories' Mark Taylor the chief computational scientist for ACME and Argonne National Laboratory's computational climate scientist, Robert Jacob.

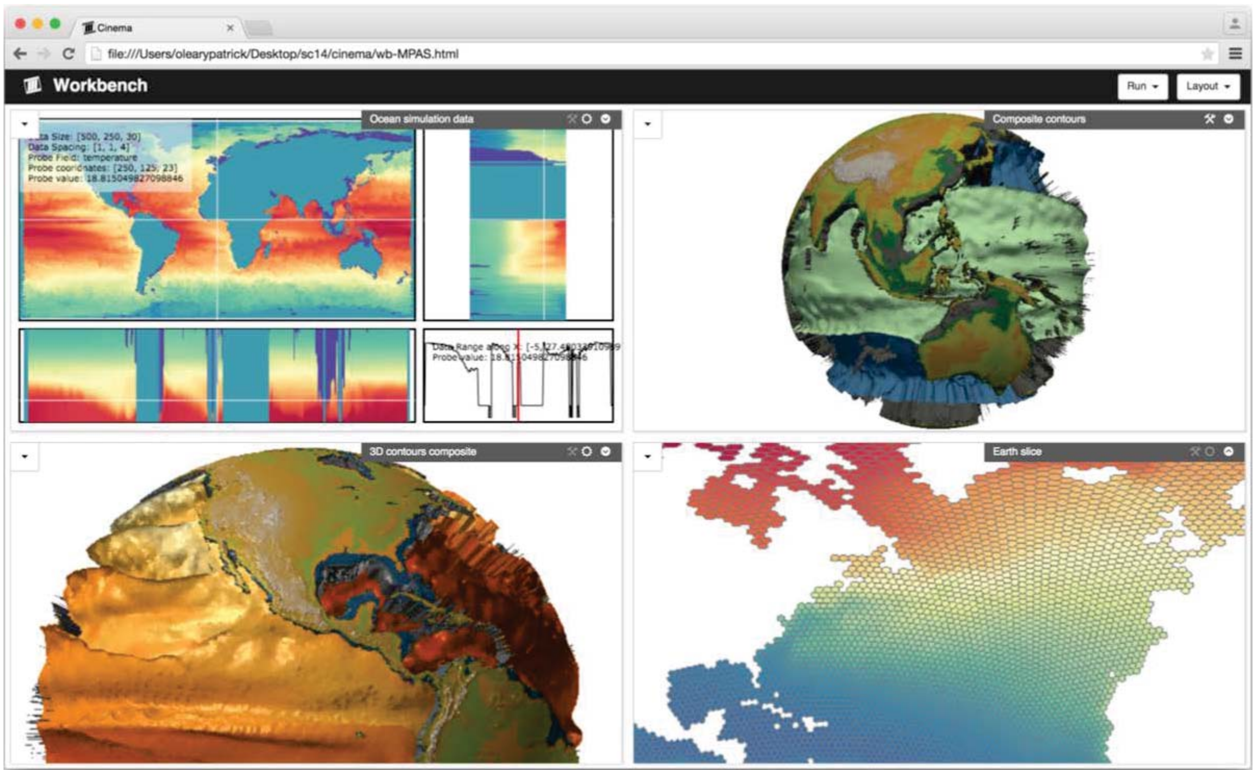

Figure 4: Cinema Workbench is demonstrating several approaches for exploration visualization of in situ produced MPAS-Ocean data products.

Accomplishments

In the first two years, we created in situ adapters for MPAS-Ocean and both CAM FD and CAM SE dynamic cores. Since May 2016, further work with the ACME codes was on hold while the stakeholders meet their program objectives and identify in situ analysis and visualization requirements.

Professional Development

The following editorial and conference work were either a direct result of this project or a collaboration result through this project.

Editorial Work

1. ISAV 2017: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC17: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Denver, Colorado, November 12, 2017 (Bauer and O'Leary).
2. ISAV 2016: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC16: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Salt Lake City, Utah, November 13, 2016 (Ayachit and O'Leary).
3. LDAV 2016: The 6th IEEE Symposium on Large Data Analysis and Visualization, held in conjunction with IEEE VIS 2016, Baltimore, Maryland. October 23, 2016 (O'Leary).
4. ISAV 2015: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC15: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Austin, Texas, November 13, 2015 (Ayachit and O'Leary).
5. LDAV 2015: The 5th IEEE Symposium on Large Data Analysis and Visualization, co-located with IEEE VIS 2015, Chicago, Illinois. October 25-26, 2015 (O'Leary).

Conference Work

1. Organizing Committee, ISAV 2017: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC17: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Denver, Colorado, November 12, 2017 (O'Leary).
2. Program Committee, ISAV 2017: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC17: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Denver, Colorado, November 12, 2017 (Bauer and O'Leary).
3. SIAM PP16 Mini-symposium MS66: In Situ Methods and Infrastructure: Answers Without All the I/O. April 15, 2016, Paris, France (O'Leary and Bauer).
4. Paper Chair, ISAV 2016: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC16: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Salt Lake City, Utah, November 13, 2016 (O'Leary).
5. Organizing Committee, ISAV 2016: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC16: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Salt Lake City, Utah, November 13, 2016(O'Leary).

6. Program Committee, ISAV 2016: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC16: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Denver, Colorado, November 12, 2017 (Ayachit and O'Leary).
7. Best Paper Committee, LDAV 2016: The 6th IEEE Symposium on Large Data Analysis and Visualization, held in conjunction with IEEE VIS 2016, Baltimore, Maryland. October 23, 2016 (O'Leary).
8. Organizing Committee, ISAV 2015: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC15: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Austin, Texas, November 13, 2015 (O'Leary).
9. Program Committee, ISAV 2015: In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization, held in conjunction with SC15: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Austin, Texas, November 13, 2015 (Ayachit and O'Leary).
10. Scientific Visualization Showcase, SC15: The International Conference on High Performance Computing, Networking, Storage, and Analysis, Austin, Texas, November 13, 2015 (Ayachit).
11. Organizing Committee, SIAM CSE 2015: SIAM Conference on Computational Science and Engineering, Salt Lake City, Utah, March 15-18, 2015 (O'Leary).
12. Student Careers Panel, SIAM CSE 2015: SIAM Conference on Computational Science and Engineering, Salt Lake City, Utah, March 15-18, 2015 (O'Leary).

Dissemination

The following disseminations were either a direct result of this project or a collaboration result through this project.

Journal Article

1. P. O'Leary, J. Ahrens, S. Jourdain, S. Wittenburg, D. H. Rogers, and M. Petersen. Cinema image-based in situ analysis and visualization of MPAS-Ocean simulations. Parallel Computing, July 2016.
2. A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. Wes Bethel. In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms, a State-of-the-art (STAR) Report. Computer Graphics Forum (Special Issue: Proceedings of EuroVis 2016), 35(3), June 2016.

Proceedings

1. P. O'Leary, S. Jhaveri, A. Chaudhary, W. Sherman, K. Martin, D. Lonie, E. Whiting, J. Money and S. McKenzie, Enhancements in VTK for Scientific Visualization in Immersive Environments, In Proceedings of the 24th IEEE Virtual Reality (VR) Conference, Los Angeles, California, pages 186–194, March 2017.

2. C. Harris, P. O'Leary, M. Grauer, A. Chaudhary, C. Kotfila, and R. O'Bara. Dynamic provisioning and execution of HPC workflows using Python. In Proceedings of the 6th Workshop on Python for High- Performance and Scientific Computing, November 2016.
3. U. Ayachit, A. Bauer, E. P. N. Duque, G. Eisenhauer, N. Ferrier, J. Gu, K. E. Jansen, B. Loring, Z. Lukic, S. Menon, D. Morozov, P. O'Leary, R. Ranjan, M. Rasquin, C. P. Stone, V. Vishwanath, G. H. Weber, B. Whitlock, M. Wolf, K. J. Wu, and E. W. Bethel. Performance Analysis, Design Considerations, and Applications of Extreme-scale In Situ Infrastructures. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'16), November 2016.
4. A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. W. Bethel. In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms, a State-of-the-art (STAR) Report. Computer Graphics Forum (Special Issue: Proceedings of EuroVis 2016), 35(3), Pages 577–597, June 2016.
5. P. O'Leary, M. Christon, S. Jourdain, C. Harris, M. Berndt, and A. Bauer. HPCCloud: A cloud/web-based simulation environment. In 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), pages 25 - 33. IEEE, December 2015.
6. U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. 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 (ISAV2015). ACM, New York, NY, USA, pp. 25-29, November 2015.
7. J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. An image- based approach to extreme scale in situ visualization and analysis. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '14), pages 424–434, Piscataway, NJ, USA, 2014. IEEE Press.
8. J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H Rogers, M. Petersen, A. Bauer, P. Fasel, F. Samsel, and B. Boeckel. In situ MPAS-Ocean image-based visualization. In Visualization & Data Analytics Showcase, International Conference for High Performance Computing, Net- working, Storage and Analysis (SC '14), 2014. http://sc14.supercomputing.org/program/ sc14-archive.

Abstracts

1. A. C. Bauer, E. Wes Bethel, B. Whitlock, and E. Lohrmann. In Situ Methods and Infrastructures: Fast Insight Through Smarter Computing. In SIAM Computational Sciences and Engineering (CSE) Mini-symposium 74, Atlanta, GA, USA, February 2017.
2. E. Wes Bethel, P. O'Leary, V. Vishwanath, and M. Wolf. In Situ Methods and Infrastructure: Answers Without All the I/O. In SIAM Parallel Processing 2016 (PP16) Mini-symposium MS66, Paris, France, April 2016.
3. M. Christon, J. Bakosi, M. Berndt, A. Bauer, A. Stagg, B. Nadiga, P. O'Leary. "Developing Hydra-TH: A Vertical, VERA-integrated Application based on the Hydra Toolkit", SIAM CSE15, March 2015.

4. A. C. Bauer, P. O'Leary, R. O'Bara, and B. Geveci. “Computational Model Builder and ParaView Catalyst: Empowering HPC Workflows.” SIAM CSE15, March 2015.
5. E. Wes Bethel (organizer), P. O’Leary, J. Clyne, V. Vishwanath, and J. Chen. In Situ Methods: Hype or Necessity? In IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC15), Austin, TX, USA, November 2015.

Talks

1. P. O’Leary, The current challenges in supporting scientific data visualization on high-end heterogeneous computer architectures, A Snapshot of Current Trends in Visualization, Guest Editors’ Theresa-Marie Rhyne and Min Chen, IEEE Computing Now (CN), February 2017.
2. A. C. Bauer. In Situ Analysis and Visualization: Background, State of the Art and Examples Using ParaView Catalyst. In Air Force Research Laboratory, U.S. Air Force, Dayton, OH, USA, July 2016.
3. A. C. Bauer. ParaView Catalyst: Effective In Situ Analysis and Visualization. In Army HPC User Group Review, Engineering Research and Development Center, U.S. Army, Vicksburg, MS, USA, August 2016.
4. P. O’Leary. Responsive large data analysis and visualization with the ParaView ecosystem. In NVIDIA GPU Technology Theater SC’15 the International Conference for High Performance Computing Networking, Storage and Analysis, Austin, TX, November 2015.
5. P. O’Leary and S. Jourdain. Is the web ready for visualization? In Workshop on Visualization Technologies at SC’15 the International Conference for High Performance Computing Networking, Storage and Analysis, Austin, TX, November 2015.
6. A. C. Bauer. ParaView Catalyst: Scalable In Situ Processing. In 2015 Air Force HPC User Forum: Visualization Session, Dayton, OH, July 2015.
7. A. C. Bauer. In Situ Analysis and Visualization for Rotor Aeromechanics Simulations. In ISAV 2015: Lightning Talk, Austin, TX, November 2015.

Presentations

1. U. Ayachit, A. C. Bauer, and P. O’Leary, Exhibited Platforms and Applications, International Conference for High Performance Computing, Networking, Storage and Analysis (SC’16), Salt Lake City, UT, November 13-18, 2016.
2. J. Jomier and P. O'Leary, Overview of Contemporary In Situ Infrastructure Tools and Architectures, 17th SIAM Conference on Parallel Processing for Scientific Computing, Paris, France, April 14, 2016.
3. E. Wes Bethel (organizer), P. O’Leary, J. Clyne, V. Vishwanath, and J. Chen, In Situ Methods: Hype or Necessity? Panel, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC’15), Austin, TX, November 20, 2015.
4. P. O’Leary and S. Jourdain, Is the Web Ready for Visualization? Keynote Presentation, VisTech '15 Workshop: Visualization Infrastructure and Systems Technology at the

International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15), Austin, TX, November 15, 2015.

5. U. Ayachit, A. C. Bauer, and P. O'Leary Exhibited Platforms and Applications, International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15), Austin, TX, November 16-19, 2015.

Outreach

- ISAV 2015, 2016 and 2017: Workshop on In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization with the workshop URL is http://vis.lbl.gov/Events/ISAV-201x(5, 6 or 7)/ (Ayachit, Bauer and O'Leary). These workshops brought together researchers, developers and practitioners from industry, academia, and government laboratories who use in situ methods in extreme-scale, high performance computing.
- SC17 tutorial proposal entitled "SENSEI Cross-Platform View of In Situ Analytics" (Bauer and O'Leary). The presenters include Bethel, Bauer, Vishwanath, Whitlock and Wolf from the SENSEI project.
- Mini-symposium at the 2017 SIAM Conference on Computational Science and Engineering, held 27 February – 3 March 2017 in Atlanta, GA (Bauer). Our mini-symposium covered topics ranging from an overview of issues motivating in situ, a survey of current tools and capabilities, examples of in situ being used in practice, and an introduction to the SENSEI in situ interface.
- An introductory full-day tutorial called "Large Scale Visualization with ParaView" was given at SC16. This tutorial included a section on using ParaView Catalyst.
- Mini-symposium for 17th SIAM Conference on Parallel Processing for Scientific Computing held in Paris, France from April 12 - 15, 2016 (O'Leary). Our mini-symposium, titled "In Situ Methods and Infrastructure: Answers Without All the I/O," was held on April 15, 2016.
- SC15 panel proposal entitled "In Situ Methods: Hype or Necessity?" (O'Leary). The panelists include Bethel, O'Leary, and Vishwanath from the SENSEI project, Jacqueline Chen (SNL-NM) to represent a "why computational science needs in situ" perspective, and John Clyne (NCAR) to represent the "why in situ is over-hyped" perspective. The panel was held on, Friday 21 November 2015, in the morning (10:30a – noon).
- An introductory half-day tutorial called "Getting Started with In Situ Analysis and Visualization Using ParaView Catalyst" was given at SC15 (Bauer). This tutorial was focused on user aspects of in situ technologies and how to leverage them efficiently in parallel. Hands-on exercises were used to show attendees how to generate Python scripts for outputting data extracts, images and a Cinema database from Catalyst.
- University at Buffalo CDSE Days (Bauer). Introductory half-day tutorials called on ParaView were given at University at Buffalo's Computational and Data-Enabled Science and Engineering Days workshop,
- SC15 and IEEE VIS 2015 ParaView Tutorials (Bauer). The tutorials were focused on new to intermediate ParaView users and provided a hands-on approach to learning how to use the ParaView GUI and Python API. Additionally, information on running ParaView efficiently in parallel was presented in each.

Collaborations

Aircraft Design - Ken Jansen (Professor at University of Colorado at Boulder) and Michel Rasquin, Post-Doctorate at Cenaero in Belgium and University of Colorado at Boulder (Ayachit, Bauer, and O'Leary).

Products

Technologies/Applications

SENSEI – The SENSEI library contains core base classes that declare the AnalysisAdaptor API which is used to interface to in situ infrastructures and implement custom analyses; the DataAdaptor API which AnalysisAdaptors use to access simulation data in a consistent way; and a number of implementations of both. For more information see our SC16 paper.

Repositories

SENSEI currently has the following publically available software repositories: https://gitlab.kitware.com/sensei/sensei.
For technology transfer, the technology developed here will also benefit the public-at-large, as we have released the code under permissive open-source licenses. Thus researchers, educators, and commercial enterprises will be able to customize the infrastructure to their particular needs.

Impact

Principle Discipline

With the advent of HPC, advanced modeling and simulation can provide faster and more detailed insights. There exists a widening gap between the ability to compute information and to store it for subsequent analysis. This gap adversely impacts science code teams, who can perform analysis only on a small fraction of the data they calculate, resulting in the substantial likelihood of lost or missed science, when results are computed but not analyzed.

Our approach is to perform as much analysis or visualization processing on data while it is still resident in memory, which is known as in situ processing. Our objective was to produce and enable the use of production-quality in situ methods and infrastructure, at scale, on DOE high-performance computing (HPC) facilities, which affect virtually all large-scale computational science efforts. To achieve this objective, we engaged in software technology research and development (R&D), in close partnerships with DOE science code teams, to produce software technologies that were shown to run efficiently at scale on DOE HPC platforms.

The focal point of our activities was to increase code portability through the development and use of a generic interface between data producers (science codes) and data consumers (in situ methods and infrastructures).

This solution directly addresses widening gap between the ability to compute information and to store it using in situ techniques. In addition, the solution addresses some critical developers need for code portability between science codes and state-of-the-art in situ methods and infrastructures.

Other Disciplines

Just as predicted in the proposal, SENSEI has been incorporated into other projects in a large number of disciplines. This technology benefits the public-at-large, as we have released our code under permissive open-source licenses across open-source communities. Thus researchers, educators, and commercial enterprises will be able to customize the platform to their particular workflow. In the paragraphs below, we have highlighted one such project.

Aircraft Design

Science Objective and Stakeholders

The design of aircraft wings is a complex process, especially when considering active flow control features such as injecting fluid into the boundary layer. Such flow control features can positively affect airfoil performance by re-attaching separated flow, reducing vibrations and virtual reshaping of the lifting surfaces. The first step in the design process is an understanding of these desired characteristics by scientists and engineers. The next step is trying to optimize airfoil performance through flow control injection placements and wing geometry. This optimization can result in several runs which each requires a significant computational capacity to capture the complex physics properly.

To gain insight promptly, in situ methods can be used to reduce the cycle time in examining different designs for favorable performance. Since the design metrics are known a priori, we can prorate initial costs for setting up the in situ analysis and visualization over the entire ensemble of runs.

PHASTA (https://github.com/PHASTA/phasta) is a general purpose CFD code developed utilized by Ken Jansen and Michel Rasquin of the University of Colorado (UC) at Boulder to examine the given workflow enhanced by HPC computing for Boeing. They demonstrated that PHASTA scaled well to up to 3 million MPI ranks on the Argonne Leadership Computing Facility's' (ALCF) Mira BG/Q supercomputer. This exploration was performed using 4 MPI ranks per core or 64 MPI ranks per node. Each node has 16 GB of memory which results in an average of 256 MB of memory available to each PHASTA MPI process. Jansen and Rasquin focused their work on simulation of a realistic tail rudder assembly.

Our science stakeholders on this are Ken Jansen from UC Boulder, who is also part of the FASTMath SciDAC Institute, and Michel Rasquin from Cenaero in Belgium and UC Boulder. One of the major goals of this project is to develop production-level capabilities for in situ analysis and visualization at the leading edge of computational capacity. This goal means that our framework needs to run at the scales of the leading edge science codes. PHASTA is a perfect science code match for this goal since it performs well at the scales that we desired to test.

Memory consumption is an important consideration for PHASTA when running on machines such as ALCF's Mira since there is only 256 MB available per MPI rank. Using PHASTA on Mira will require our framework to both scale and use memory efficiently.

PHASTA was one of the earliest codes to be instrumented with the Catalyst in situ infrastructure. Because of this, it was relatively straightforward to modify PHASTA to use the SENSEI generic in situ interface API.

We wrote the original data adaptor before we developed any of the zero-copy infrastructure in VTK. While the memory for PHASTA's scalar arrays could be used directly in VTK's data arrays for the initial integration of the data adaptor, the velocity and point coordinate vector arrays had to be deep copied due to PHASTA using a structure-of-arrays (SOA) memory layout. With the zero-copy functionality within VTK, we modified the data adaptor to reduce the in situ memory consumption. To run efficiently on Mira, we also built an edition of Catalyst enabling the slice analysis adaptor to reduce library size and thus memory footprint. The slice analysis adaptor computes a slice through the domain and then renders an image of the generated slice. The resulting static executable size was 153 MB while the executable size without Catalyst was 87 MB. Finally, we have reduced the file system introspections during Catalyst initialization by performing the file IO on the first MPI process and broadcasting that information to the other processes.

Technology Transfer

As mentioned earlier, the technology developed here also benefits the public-at-large, as we have released our code under permissive open-source licenses across open-source communities. Thus researchers, educators, and commercial enterprises will be able to customize the platform to their particular workflow. Further, since the platform provides generic capabilities for advanced modeling and simulation workflows on HPC resources, it can be used to address most problems that can benefit from advanced modeling and simulation such as materials science and even healthcare.

Conclusions

The primary challenge motivating this project is the widening gap between the ability to compute information and to store it for subsequent analysis. This gap adversely impacts science code teams, who can perform analysis only on a small fraction of the data they calculate, resulting in the substantial likelihood of lost or missed science, when results are computed but not analyzed. Our approach is to perform as much analysis or visualization processing on data while it is still resident in memory, which is known as in situ processing. The idea in situ processing was not new at the time of the start of this effort in 2014, but efforts in that space were largely ad hoc, and there was no concerted effort within the research community that aimed to foster production-quality software tools suitable for use by Department of Energy (DOE) science projects. Our objective was to produce and enable the use of production-quality in situ methods and infrastructure, at scale, on DOE high-performance computing (HPC) facilities, though we expected to have an impact beyond DOE due to the widespread nature of the challenges, which affect virtually all large-scale computational science efforts. To achieve this objective, we engaged in software technology research and development (R&D), in close partnerships with DOE science code teams, to produce software technologies that were shown to run efficiently at scale on DOE HPC platforms.

The main focal points of our activities were as follows:

- Show the viability of in situ methods and infrastructures, at scale, on contemporary DOE HPC platforms within the context of several DOE science exemplars. Our approach consisted of annual campaigns targeting runs, at increasing scale, on multiple DOE HPC platforms. We used a combination of mini-applications and science codes and focused on a limited set of design and execution patterns to limit problem scope.
- To increase code portability through the development and use of a generic interface between data producers (science codes) and data consumers (in situ methods and infrastructures).
- Engage in close partnerships with DOE science code teams, who provided scientific-based motivation for the software technology R&D and whom also benefited through the integration of this software to utilize associated in situ methods and infrastructures.
- Through outreach, publications, and presentations, foster the coordination and formation of a community of in situ processing developers, researchers, and practitioners.
- Produce and release software artifacts, consisting of the generic in situ interface, methods, and infrastructures.

Over the course of the project, we have been highly productive regarding generating publications:

- Seven peer-reviewed journal articles (IEEE Visualization; Computer Graphics Forum; Parallel Computing; Computational Astrophysics and Cosmology);
- Nineteen peer-reviewed conference papers (SCXX, HPDC, IEEE Cluster, Grid, and Cloud Computing; eScience; Parallel CFD; ALENEX; IPDPS; and many others);
- Eight peer-reviewed workshop papers; and
- Four peer-reviewed panel abstracts (LDAV, SCXX, SIAM PP16, SIAM CSE 2017).

The team has also been highly active in outreach and fostering interaction within the community of in situ researchers and practitioners outside this project. We launched a new workshop at the SC annual conference series, In Situ Infrastructures for Enabling Extreme Scale Analysis and Visualization (ISAV), held in 2015 and 2016, and currently accepted for the 2017 program. We have also organized multiple mini-symposia at SIAM meetings, presented tutorials at SC in 2015 and 2016, engaged in direct outreach on various topics related to in situ technologies.

Recommendations

Kitware, Inc. benefited from participation in DOE's Office of Science, Advanced Scientific Computing Research, PI Meeting where we presented our project and visited with DOE scientists and engineers about their work and issues. It is our recommendation and hopes that all future projects be gifted with program managers that identify appropriate like opportunities.