Journal Title
XX(X):1–15
©The Author(s) 2024
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

The ECP ALPINE project: In Situ and Post Hoc Visualization Infrastructure and Analysis Capabilities for Exascale

James Ahrens[1], Marco Arienti[6], Utkarsh Ayachit[17], Janine Bennett[6], Roba Binyahib[12], Ayan Biswas[1], Peer-Timo Bremer[4], Eric Brugger[4], Roxana Bujack[1], Hamish Carr[9], Jieyang Chen[5], Hank Childs[7], Soumya Dutta[13], Abdelilah Essiari[3], Berk Geveci[10], Cyrus Harrison[4], Subhashis Hazarika[15], Megan Hickman Fulp[18], Petar Hristov[9], Xuan Huang[8], Joseph Insley[2], Yuya Kawakami[7], Chloe Keilers[1], James Kress[19], Matthew Larsen[11], Dan Lipsa[10], Meghanto Majumder[7], Nicole Marsaglia[4], Victor Mateevitsi[2], Valerio Pascucci[8], John Patchett[1], Saumil Patel[2], Steve Petruzza[8, 16], David Pugmire[5], Silvio Rizzi[2], David H. Rogers[1], Oliver Rübel[3], Jorge Salinas[6], Sudhanshu Sane[8], Sergei Shudler[14], Alexandra Stewart[1], Karen Tsai[1], Terece L Turton[1], Will Usher[11], Zhe Wang[5], Gunther H. Weber[3], Corey Wetterer-Nelson[10], Jonathan Woodring[1], Abhishek Yenpure[10]

Abstract

A significant challenge on an exascale computer is the speed at which we compute results exceeds by many orders of magnitude the speed at which we save these results. Therefore the Exascale Computing Project (ECP) ALPINE project focuses on providing exascale-ready visualization solutions including in situ processing. In situ visualization and analysis runs as the simulation is run, on simulations results are they are generated avoiding the need to save entire simulations to storage for later analysis. The ALPINE project made post hoc visualization tools, ParaView and VisIt, exascale ready and developed in situ algorithms and infrastructures. The suite of ALPINE algorithms developed under ECP includes novel approaches to enable automated data analysis and visualization to focus on the most important aspects of the simulation. Many of the algorithms also provide data reduction benefits to meet the I/O challenges at exascale. ALPINE developed a new lightweight in situ infrastructure, Ascent.

Introduction

Prior to the exascale era, typical visualization tasks and analysis used post hoc visualization workflows leveraging a visualization application such as ParaView Ahrens et al. (2005) or VisIt Childs et al. (2012). Post hoc workflows visualize simulation output data that was previously saved during simulation execution. Having reached the exascale regime, scientific simulations can now produce terabytes of data in every time step. Recent advances in I/O and storage capabilities have not kept up with the increases in compute power. Given these challenges, in situ approaches are a viable and necessary solution to meeting the needs of high performance computing applications. In situ approaches are run during a simulation, possibly at each time step, processing simulation outputs as they are generated. In situ analysis and visualization approaches can be used to downselect and reduce data, identify features of interest, produce visualizations, and generate smaller extracts that can be used in post hoc workflows. In situ infrastructures provide the necessary application and systems interfaces to support in situ workflows.

The contributions of the Exascale Computing Project's (ECP) ALPINE project were:

1. Made post hoc visualization tools exascale ready
2. Developed Exascale visualization and analysis algorithms that will be critical for ECP Applications.
3. Developed an Exascale-capable infrastructure for the development of in situ algorithms and deployment into existing applications, libraries, and tools.
4. Integrated other ECP Software Technology data and visualization, and programming model products into our infrastructure.
5. Integrated our algorithms and infrastructure into ECP Applications.

[1]Los Alamos National Laboratory, [2]Argonne National Laboratory, [3]Lawrence Berkeley National Laboratory, [4]Lawrence Livermore National Laboratory, [5]Oak Ridge National Laboratory, [6]Sandia National Laboratories, [7]University of Oregon, [8]University of Utah, [9]University of Leeds, [10]Kitware, Inc., [11]Luminary Cloud, [12]Intel, [13]Indian Institute of Technology Kanpur, [14]e.solutions GmbH, [15]SRI International, [16]Utah State University [17]NVIDIA, [18]Clemson University, [19]King Abdullah University of Science and Technology

Corresponding author:
Terece L. Turton, Los Alamos National Laboratory LANL MS-B256, PO Box 1663, Los Alamos, NM 87545-0001
Email: tlturton@lanl.gov

This paper is structured to describe these contributions with sections on: in situ infrastructure, in situ algorithms, software integrations and application integrations.

ALPINE Infrastructure

A key challenge of ECP was achieving high performance, portable, thread-based parallelism on graphical processing units (GPUs). The Visualization Toolkit (VTK) is a open source visualization library that offers full-featured collection of visualization and analysis filters. VTK algorithms are used by ParaView and VisIt. A distributed memory version of VTK was previously developed to run scalably on supercomputers offering across-node parallelism. The VTK-m Moreland et al. (2016) project, a companion ECP project, developed portable multi-threaded implementations of key VTK visualization and analysis algorithms for on-node parallelism. ALPINE infrastructure has developed a layer on top of the VTK-m library for cross-platform portability and performance. This layer is where all ALPINE algorithms are implemented, and it is deployed in ParaView, Catalyst, VisIt, and Ascent. Thus all development effort by ALPINE is available in all of the tools and, by leveraging VTK-m, addresses issues with portability and many-core architectures. (Note: post-ECP, VTK-m is now available as Viskores Moreland et al. (2024).)

ParaView

ParaView 5.11.1, the open source platform for scientific visualization, was deployed on the Frontier supercomputer at the Oak Ridge Leadership Computing Facility (OLCF), enabling analysis and rendering workflows which take advantage of the exascale computing capabilities of the facility.

To enable ParaView to utilize GPUs on the exascale machines, a new set of "accelerated filters" were implemented. These filters serve as wrappers over VTK-m's filters. These accelerated filters readily use exascale hardware and have been demonstrated to be performant. The accelerated filters are available in ParaView as plugins which can be loaded on demand, and in case of failures, a fallback has been provided to use the traditional (VTK) filters. Additionally, these filters also handle all the necessary conversions between the ParaView and VTK-m data models without unnecessary data movement (zero-copy). Deployment was enabled by new developments in Spack under the ECP DAV-SDK (software develoment kit) project.

Analysis and visualization workflows were validated on massive datasets, such as that shown in Figure 1, as well as synthetic structured grid datasets composed of over 4.4 trillion elements, taking up over 16.4TB per timestep on disk. ParaView was able to take advantage of the considerable GPU resources on Frontier for accelerated analysis filters employing VTK-m.

VisIt

VisIt is an interactive, scalable, distributed visualization and analysis tool. VisIt uses the Visualization Toolkit (VTK) to provide much of its visualization and analysis capabilities. This functionality is encapsulated in a filter architecture.

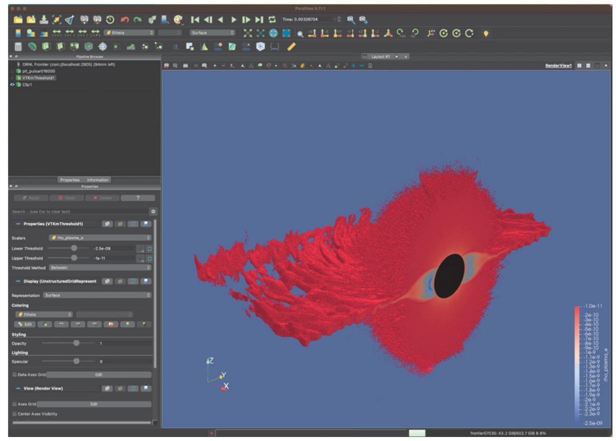

Figure 1. A single timestep from a simulation of a pulsar conducted in WarpX is rendered in ParaView running remotely on Frontier at OLCF. This dataset is composed of a 5.75B element AMR mesh totaling 1.16TB of data. This visualization takes advantage of GPU accelerated VTK-m analysis filters employing 128 nodes and a total of 512 GPUs. This data is courtesy of Revathi Jambunathan at Lawrence Berkeley National Lab.

VisIt uses distributed memory parallelism using MPI to scale its functionality to the largest DOE leadership class systems. The main thrust of the VisIt effort in ECP ALPINE was to leverage on node parallelism using VTK-m, culminating in the release of VisIt 3.3.3 on Frontier. The user can now specify, either through the Python scripting interface or the graphical user interface, whether to use the traditional VTK filters or to use VTK-m filters when possible.

VisIt's internal filters were enhanced to support using either VTK or VTK-m. When VTK-m is enabled and a filter supports VTK-m then it converts the dataset to VTK-m if necessary and then uses the VTK-m filter. The conversion is done using zero-copy constructs wherever possible to minimize data duplication. Several of the most heavily used filters were converted to use either VTK or VTK-m including Contour, Slice, Clip, Isovolume and Threshold.

VisIt's Spack package was enhanced to support building VTK-m with the Kokkos backend using HIP. Additionally, several other optional VisIt dependencies were added to VisIt's Spack package including Conduit Harrison et al. (2022) and MFEM Anderson et al. (2021).

To demonstrate running at scale using the AMD APUs on Frontier, VisIt was used to generate an image, Figure 3, from a 2048 domain WarpX calculation with 70 billion zones. VisIt was run on 512 nodes using 2048 APUs.

Ascent

Developed under the ALPINE project, Ascent is a flyweight in situ visualization and analysis library for multi-physics simulations targeting current and next-generation HPC architectures. It was designed and built from the start to leverage GPUs for on-node parallelism. Ascent productized and expanded the flyweight software architecture prototype of the Strawman Larsen et al. (2015) Larsen et al. (2017) in situ visualization proxy.

Ascent aims to be easy to use, providing three main use cases: making pictures, transforming data, and capturing

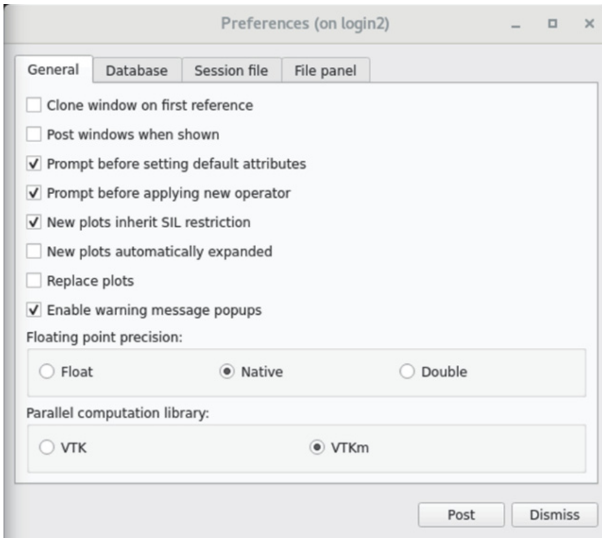

Figure 2. Selecting the VTK-m backend in VisIt.

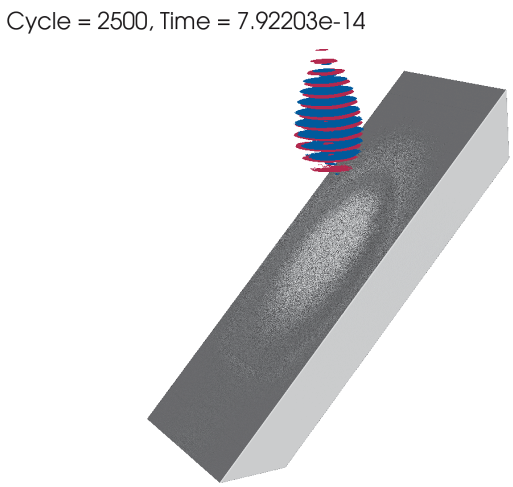

Figure 3. Visualization from the 70 billion cell WarpX Gordon Bell-winning simulation Fedeli et al. (2022) visualized with 2048 GCDs on Frontier using VisIt.

data. To pass data to Ascent, Ascent leverages Conduit Blueprint to intuitively describe simulation mesh data. Ascent was the first production infrastructure to demonstrate Conduit Blueprint as a viable strategy for sharing simulation mesh data in situ. The Ascent team worked closely with ECP Co-Design Centers to create easy paths to publish simulation mesh data to Ascent from codes using AMReX Zhang et al. (2019) or MFEM.

Ascent supports the most common visualization and analysis operations, provides infrastructure to integrate custom analysis, and creates several types of extracts including HDF5 The HDF Group files and Cinema databases Ahrens et al. (2015). Ascent uses Conduit to provide C, C++, Python, YAML, and Fortran APIs to describe which visualization actions to execute. Ascent requires minimal dependencies resulting in lower memory requirements than other current tools, resulting in a flyweight design with a small memory footprint, while leveraging libraries that provide parallel performance.

To achieve performance and portability, Ascent leverages the VTK-m library and RAJA Beckingsale et al. (2019) for on-node parallelism, and MPI (Message Passing Interface) for distributed-memory coordination. VTK-m provides a suite of visualization and analysis algorithms, as well as zero-copy capabilities and the ability to pass device-pointers, allowing for efficient exploitation of shared resources. Ascent was also a platform for research into in situ triggers Larsen et al. (2018) Larsen et al. (2021) Lawson et al. (2021), which provide flexibility to adapt visualization actions and help address a priori constraints that can limit batch use of in situ tools.

Catalyst

Under ECP ALPINE, the Catalyst Ayachit et al. (2021) in situ analysis and visualization platform was expanded and matured to meet the requirements of advanced exascale simulation workflows.

The standalone Catalyst 2.0 API leverages Conduit Blueprint to describe simulation data and manage its transmission to runtime selectable backends which execute analysis and visualization workflows. By using the same library as Ascent to manage data description and transmission, simulation applications can use both in situ libraries with little changes to their codebase, significantly increasing the surface area for both Catalyst and Ascent across the exascale simulation workflow ecosystem. This benefit was showcased by the rapid integration of a Catalyst in situ analysis adapter to MFiX-Exa, a massively parallel computational fluid dynamics–discrete element model (CFD-DEM) code, to study multiphase flows Musser et al. (2022). Results of that integration are shown in the ALPINE Integrations Highlights section later in this article.

Further increasing the usability and applicability of Catalyst, language support for Python and Fortran were added to the Catalyst 2.0 API under ECP ALPINE. This effort provided bindings so that the Catalyst 2.0 API can be called from Python and Fortran based simulation codes directly. This development leveraged the existing Conduit bindings for the two languages.

One of the key advantages of Catalyst is runtime selectable backends. Here, a user can decide at simulation startup whether to use, for instance, the ParaView Catalyst backend for full featured leadership class analysis and visualization workflows, or the ADIOS Catalyst backend, targeting in-transit workflows for asynchronous analysis activities, or the new Ascent backend developed under ECP ALPINE, for direct access to GPU accelerated Ascent tools. Because Catalyst 2.0 utilizes Conduit in similar ways to Ascent, it was natural to expose Ascent as a backend for Catalyst, allowing existing Catalyst users to easily employ Ascent in situ workflows.

Task-based Composable Workflows

In the realm of in situ processing, where analysis routines seamlessly integrate with simulation code stacks, a notable distinction arises. Unlike simulation code, analysis and visualization routines are generally applicable across a broad spectrum of applications. However, complications emerge when different simulation codes operate on varied architectures or runtimes, leading to the constant need to tailor analysis code to specific hardware. A multi-runtime abstraction layer called BabelFlow Petruzza et al. (2018) was introduced to address these challenges, offering developers a straightforward dataflow-based interface for the implementation of parallel algorithms. By utilizing task graphs, BabelFlow explicitly delineates parallel execution sections of the algorithm and their interrelations.

This framework has been integrated into Ascent to allow the implementation of task based analysis and visualization algorithms and has been extended to support the composition of dataflow graphs into more complex workflows. This extension, called LegoFlow Shudler et al. (2021), currently provides task based in situ workflows for: (i) a distributed rendering and image compositing using Devil Ray DevilRay and VTK-h VTK-h (described in Shudler et al. (2021)); and (ii) a merge tree based segmentation and feature statistics computation. The merge tree based analysis segments the domain into features according to threshold values (i.e., level sets). This kind of segmentation has been proven to be useful in a number of scenarios, such as extracting extinction regions in turbulent combustion simulations, or identifying and tracking eddies in the oceans. We have extended the merge tree computation workflow Petruzza et al. (2018) to compute statistics of the features extracted using a streaming statistics library Shudler and Bremer (2022).

ALPINE Algorithms

The development of innovative algorithms to support the needs of exascale applications was an important facet of ALPINE's contributions to ECP. Algorithm development generally began with a basic Python or C++ prototype. ECP science application partners shared early datasets of interest which were used for prototype testing and to gauge the impact of algorithm for potential use. In order for algorithms to be accessed in both post hoc and in situ infrastructures, final algorithm productization required converting the algorithm to a VTK-m filter with associated unit testing.

- Topological analysis: These methods are used to detect features in the data and adaptively steer visualizations. For example, contour trees can identify the most significant isosurfaces in complex simulations and then the resulting visualizations can use these isosurfaces.
- Adaptive sampling: These methods can be used to guide visualizations and extracts to the most important parts of the simulation, significantly reducing I/O.
- Statistical feature detection: These methods use distribution-based approaches and statistical similarity measures to identify and isolate features of interest. Significant data reduction is possible by only saving the statistical representations of the data.
- Lagrangian flow analysis: This method is used to analyze fluid flow, allowing more efficient and complete tracking of particles over time. It can save time-varying vector field data with higher accuracy and less storage than the traditional approaches
- Optimal viewpoint selection: These metrics can be used to automate visualization decisions in situ, minimizing visualizations written to disk.
- Rotational invariant pattern detection algorithm.

Topological analysis

Visualization increasingly requires analytic tools for data beyond human comprehension: tools such as the *contour tree*, *Reeb graph* and *merge tree* which summarize the development of features in the data set as the isovalue varies are therefore of prime interest. However, the application of these tools has been limited by the scalability of often serial algorithms, in particular the standard serial algorithm Carr et al. (2003) for merge and contour trees.

Our goal in ECP ALPINE was to use the contour tree for selection of isosurfaces on exascale machines, see Figure 4. This required algorithms using both on-node (shared memory) parallelism and multi-node (distributed) parallelism. We achieved this through a hybrid algorithm, using data parallel primitives, VTK-m and DIY Morozov and Peterka (2016) for portability.

To do so, we introduced (data-) parallel peak pruning (PPP) Carr et al. (2021), exploiting parallel-friendly properties of monotone paths instead of the serializing properties of contours previously used Carr et al. (2003). However, computing the contour tree alone is insufficient, as it captures only critical points where topology changes, where analysis requires further information about "regular" points where topology is invariant. We therefore extended this algorithm to compute the fully-augmented contour tree Carr et al. (2022a), based on data-parallel *hyperstructures* for acceleration. With these, we were able to implement data-parallel data analysis using the contour tree and tie it into the Cinema database for single-node visualization Hristov et al. (2020).

Based on an efficient single-node contour tree algorithm, we then developed a distributed, hierarchical representation of the contour tree Carr et al. (2022b), based on the hyperstructure used in shared-memory. This in turn allowed us to extend analysis and visualization tools to hybrid distributed parallelism, supporting geometric computations, branch decomposition and selection of the most relevant contours. Finally, we coupled our contour-tree based analysis using Ascent to the WarpX simulation code and ran tests, see Figure 5.

Both, the single node contour tree algorithm as well as the distributed version are available to anyone through VTK-m.

Adaptive Sampling

Sampling is an in situ data reduction approach for scalar datasets generated by large-scale scientific simulations. Under ECP, ALPINE developed several data-driven sampling methods. The most generic sampling method essentially analyzes the scalar data distribution and local smoothness property of data to automatically assign *importance* to the scalar values. Points in the field are accepted (i.e, kept for post hoc analysis) or rejected (i.e., removed during in situ processing) based on their importance. Typically, important features are the rarer events. Thus the automated sampling approach assigns higher importance to the low probability scalars and lower importance to the higher frequency scalars.

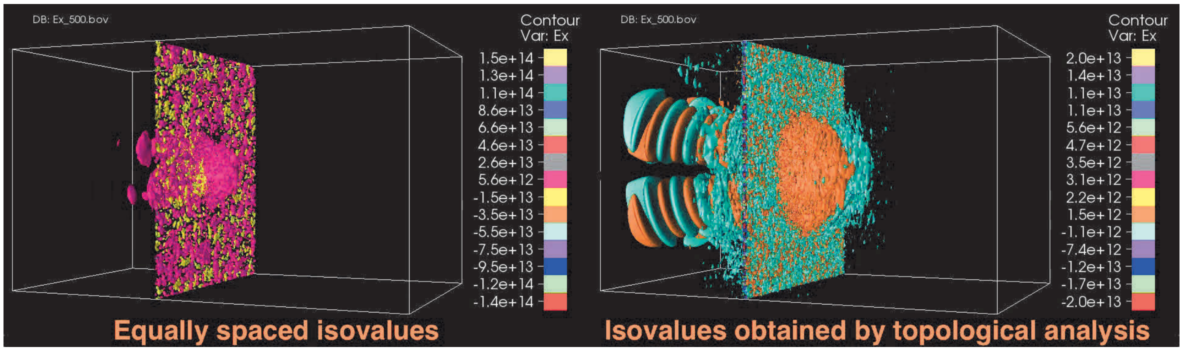

Figure 4. Comparing contours for equally spaced isovalues to contours selected using topological analysis via the contour tree for a Warp simulation. This early illustrative was example created via post hoc analysis of a Warp simulation.

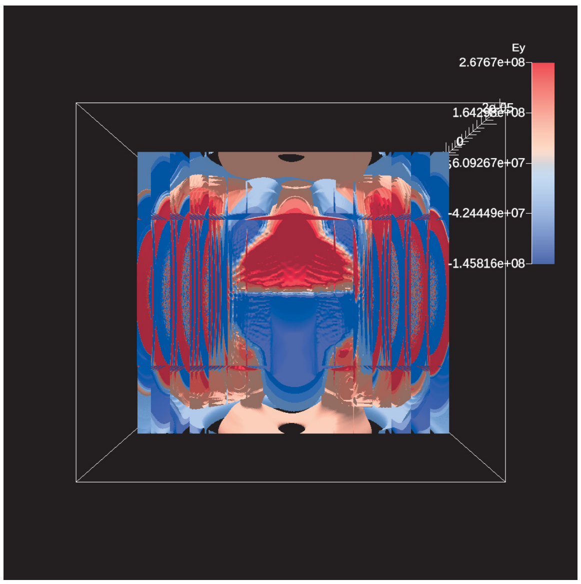

Figure 5. Small scale run of contour selection using topological data analysis via the contour tree. This image was created using a WarpX simulation instrumented with Ascent. This is a 32 node run on Frontier using 256 MPI ranks.

The other aspect of importance is based on local smoothness or local gradient information. High gradient regions often are of high importance to the domain experts as they can indicate feature boundaries or regions of high turbulence or mixing. The high gradient sampling scheme exploits local smoothness to assign higher importance to high gradient regions alongside the previously mentioned value-based importance. An example of this sampling scheme is shown in Figure 6. Figure 6a shows the volume rendering of density field from Nyx simulation and Figure 6b shows the particles remaining after applying the data-driven sampling scheme. As can be seen in those two figures, the sampled particles follow the structures of the density field quite closely.

The sampling algorithm is available through Ascent as a VTK-m filter. Two versions of this algorithm are available: a histogram-based sampling using importance and a histogram+gradient-based sampling. The histogram-based version emphasizes the scalar value distribution alone, whereas the histogram+gradient-based version considers

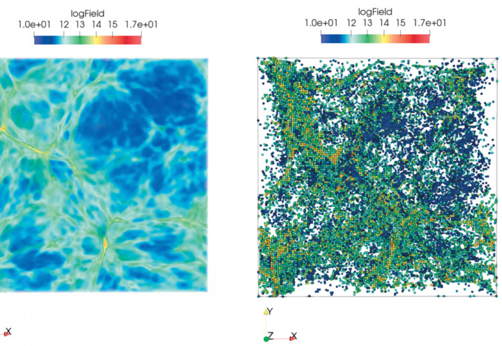

Figure 6. Left: the density field from Nyx simulation; right: the sampled particles data from the density field.

the joint distribution of both the scalar values as well as the gradient magnitude values. Including the scalar value distribution helps preserve the low-frequency regions of the data, while gradient magnitudes emphasize the smoothness of the data in those regions. Therefore, the histogram+gradient-based version is generally better at retaining important features of the data than just the histogram-based version of sampling algorithm. The interested reader is directed to Biswas et al., Biswas et al. (2022, 2021) for further information.

Lagrangian Flow Analysis

Lagrangian analysis is an in situ data reduction operator used for time-dependent vector field data generated by a simulation code. With the objective of storing/representing fluid dynamics data in its Lagrangian representation, the Lagrangian analysis functionality is implemented as a VTK-m filter. The filter operates by placing seeds and calculating the corresponding particle trajectories in the flow volume. These particle trajectories encode the underlying behavior of the flow field. Calculating and extracting a Lagrangian representation of a flow field offers significantly improved accuracy-storage propositions for time-dependent flow visualization compared to the traditional (Eulerian) method. Thus, the Lagrangian analysis filter enables data reduction of large vector fields while maintaining high data integrity. Computing a Lagrangian representation using in situ processing and storing a reduced flow map representation of the vector field can potentially address the shortcomings of the traditional approach.

The VTK-m Lagrangian flow analysis filter produces flow maps when provided with time-varying vector field data and manages particles on a per rank basis. The flow maps themselves consist of the start locations and displacement of each particle over several simulation iterations, thus capturing the behavior of the particle over an interval of time. To maintain domain coverage, particles are reset to their initial start position after each interval. The flow maps can be interpolated directly to generate new particle trajectories accurately. The efficacy of the approach has been demonstrated on multiple computational fluid dynamics applications including cosmology, seismology, and hydrodynamics. The interested reader is invited to peruse Sane et al. (2018), Sane et al. (2021a), Sane et al. (2021b), Sane and Childs (2022), Sane et al. (2022) for example use cases.

Optimal Viewpoint Selection

Optimal viewpoint selection is an in situ algorithm for automating camera placement for in situ visualization of multi-physics HPC simulations. The algorithm operates on mesh data and uses Viewpoint Quality (VQ) metrics to evaluate how much insight a camera position provides. Typically, VQ metrics analyze some visible aspect of the visible data, such as the geometry or field data. In order to determine which VQ metrics best represent choices a domain scientist would make, a user study (complying with institutional requirements for human subject research) with large data analysis and visualization experts was performed and resulted in a new, entropy-based VQ metric that best predicts user preference Marsaglia et al. (2021). The entropy-based VQ metric is a combination of three entropy calculations: entropy of the visible field data, entropy of the visible depths (measured from the camera to the geometry), and the entropy of the visible shading values.

Optimal viewpoint selection was implemented as a filter in the Ascent in situ visualization and analysis framework. The VQ metrics were written using VTK-m to guarantee shared-memory performance and portability, as well as MPI for efficient distributed-memory parallelism Marsaglia et al. (2022b). Optimal viewpoint selection can be useful for exploratory purposes when there is no a priori knowledge of the simulation, it can also be used as a trigger when the simulation has changed Marsaglia et al. (2022a). However, more importantly, the optimal viewpoint selection minimizes the amount of data written to disk, reducing a large-scale simulation time step to a single, insightful image.

Statistical Feature Detection

The Statistical feature detection algorithm processes three-dimensional (3D) particle fields in situ and transforms the data into a feature similarity field, which is stored to disk for further post hoc analysis. The current version of the algorithm works on a particle field; however, the algorithm can be easily applied to any regular-grid scalar data with minor modifications. Starting with analyzing data in situ and detecting features of interest to the user, the algorithm then outputs a statistically summarized data set that is significantly smaller in size compared to the raw particle data. The summarized data can be analyzed interactively in post hoc analysis for further feature analysis. This algorithm follows the feature-driven data reduction paradigm to achieve significant data reduction while preserving important information so that post hoc analysis can be done on the reduced data.

The algorithm works on an unstructured particle field and a feature is represented as a statistical probability distribution. Representing the feature in the form of a distribution allows the application scientists to specify a descriptor of the features of interest without needing to precisely define it. In many application domains, a precise description of a feature is not readily available due to the complexity of the scientific data. A statistical technique is a flexible solution for feature detection. An interactive user interface can be used where the users can move a cube object freely inside the data and put it in a region where they are interested. Next, a distribution representation (currently Gaussian distribution is used, but any other distribution model can be used) is created from the data points within that selected cube region and is used as the target feature descriptor.

The ECP use case was the MFIX-Exa CFD-DEM code. The feature of interest is an area of low density or a bubble. For this particle-based code, the algorithm takes a particle field as input and first transforms it into a regular grid particle density field. The density field is passed through a 3D super voxel generating algorithm, called Simple Linear Iterative Clustering (SLIC) that produces super voxels from the particle density field. A Gaussian distribution is modeled for each super voxel. Finally, a distribution similarity measure is used to compute a statistical similarity field between each super voxel distribution and the user-provided target feature distribution. These steps can be seen in Figure 7. The interested reader is directed to Dutta et al. Dutta et al. (2022a) and Dutta et al., Dutta et al. (2022b) for details.

Rotational Invariant Pattern Detection

Pattern detection can be used to identify features in a simulation in situ to reduce the amount of data that needs to be written to disk. For simulations where physically meaningful patterns are already known, the orientation of the pattern may not be known a priori. Pattern detection can be unnecessarily slowed if the pattern detection algorithm must search for all possible rotated copies of a pattern template. Therefore, rotation invariance is a critical requirement. Moment invariants can achieve rotation invariance without the need for point to point correlations, which are difficult to generate in smooth fields Bujack and Hagen (2017); Bujack et al. (2022).

The rotational invariance feature detection algorithm can take either scalar or vector fields and requires a pattern template as input. An example using the same MFIX-Exa dataset defined the search pattern to be a density boundary between a high density and low density field. The first step of the VTK-based filter computes the moments while the second step performs a normalization based on the given pattern that makes them invariants. Then, the third step computes the similarity between each part of the simulation and the template. Figure 8 shows the original

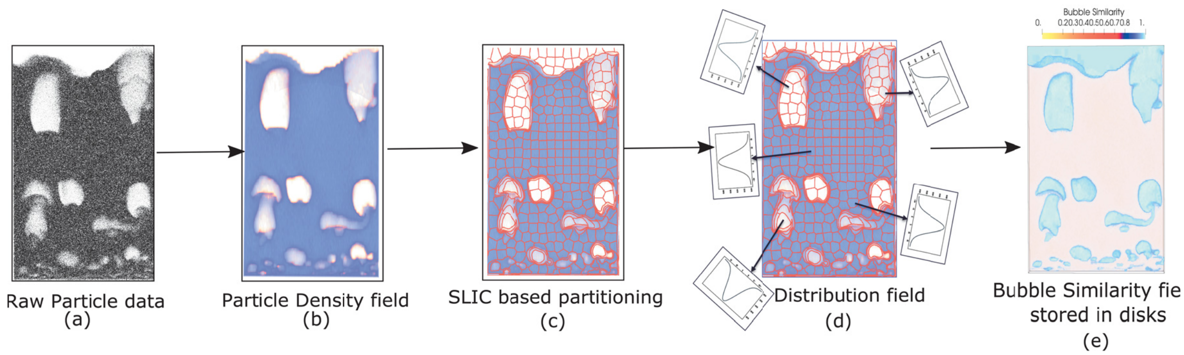

Figure 7. The steps of the in situ statistical feature detection algorithm from the raw data to the similarity field.

pattern template and data along with the bubbles identified with this algorithm. The interested reader can find more details in Tsai et al., Bujack et al. (2018); Tsai et al. (2020).

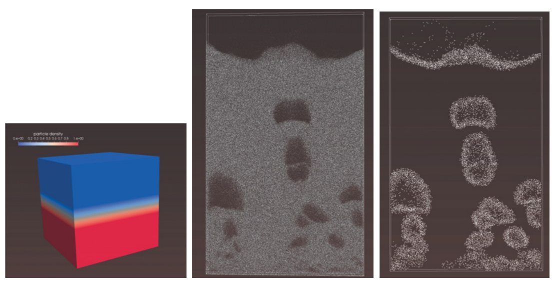

Figure 8. Left: the density boundary as the search pattern; middle: the original particle dataset; right: the identified bubbles in the data.

ALPINE Software Technology Integration Highlights

ECP's data and visualization (DAV) portfolio is a software stack of products designed to support data management, data analysis, and visualization needs at exascale. With the emphasis on interoperability, ALPINE infrastructures can be used to link client applications to capabilities across the DAV portfolio and other ECP capabilities. In particular, ALPINE relies on VTK-m for cross-platform portability and visualization filters. By integrating ECP co-design codes such as AMReX Zhang et al. (2019) into ALPINE infrastructures, AMReX-based applications can easily access ALPINE capabilities. All ALPINE infrastructures support HDF5 for I/O and, through HDF5 The HDF Group, access to the zfp Lindstrom (2014) and SZ Di and Cappello (2016) compressors. Cinema databases Ahrens et al. (2015) can be exported in situ from ALPINE infrastructures to support post hoc visualization and analysis workflows. Ascent, in addition to VTK-m Moreland et al. (2016) for portability, includes a RAJA Beckingsale et al. (2019) backend. The MFEM Anderson et al. (2021) high-order finite element library has also been integrated into Ascent. Through the DAV Software Development Kit DAV SDK, all ALPINE capabilities are available in the Extreme-scale Scientific Software Stack (E4S) Heroux et al. (2023) for post-ECP sustainability.

ALPINE Application Integration Highlights

The success of our project is demonstrated by the integration of our in situ algorithms and infrastructure into ECP applications. In this section, we highlight our integration with the Combustion-Pele, WarpX, and MFIX-Exa projects.

Integration of Combustion-Pele with Ascent and ExaLearn

An anomaly can be loosely defined as an occurrence of something that is "abnormal", "atypical" or "unexpected". Here, we have implemented a methodology that is centered on analyzing high-order joint moments in multi-variate combustion datasets, and then applied it to the problem of identifying the onset of autoignition in a combustible mixture in situ. The methodology is based on the co-kurtosis algorithm by Aditya et al. (Aditya et al. 2019) for calculation and analysis of fourth-order joint moments. Kurtosis is a measure of either existing outliers (for the sample kurtosis) or of the propensity to produce outliers (for the kurtosis of a probability distribution; Westfall (2014)). The integration between the co-kurtosis calculation, implemented via ExaLearn Alexander et al. (2021) into the exascale code for reacting flows Pele Henry de Frahan et al. (2024) was powered by ALPINE Ascent, the flyweight visualization and analysis infrastructure for multi-physics HPC simulations. Thanks to the combination of the adaptive mesh refinement granularity in Pele with the statistical outlier detection capability of co-kurtosis, the method demonstrated a considerable speed-up compared to traditional post-processing techniques when tested for the identification of ignition kernels from the injection of a Diesel-like fuel in air (Borghesi et al. 2018). Using all AMR levels and all chemical species, the entire process of identification was shown to take only 2% per solver time step in its target run (2.4 Trillion degrees of freedom) on 56,800 GPUs, thus demonstrating its in situ effectiveness. The metrics generated at runtime for AMR levels 3 to 6 are shown in Figure 9, where each AMR block (each small cube shown) is comprised of between 16^3 and 64^3 cells.

WarpX Visualization Pipeline

WarpX is an award winning particle-in-cell simulation code that studies advanced particle acceleration in laser-driven plasma wakefields Fedeli et al. (2022) in order to advance

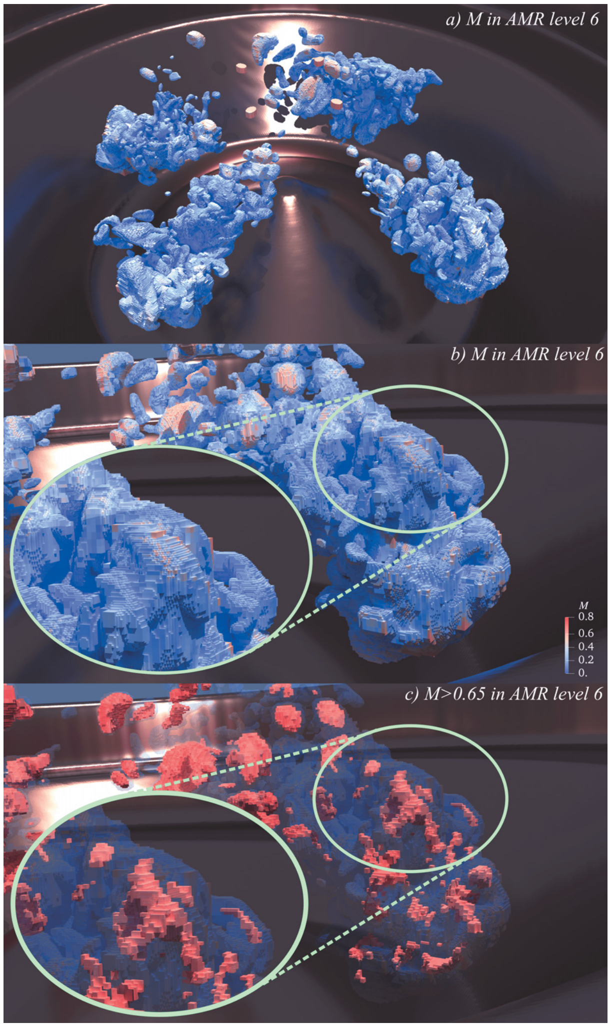

Figure 9. Simulation of direct injection of four jets of prevaporized n-dodecane fuel-air mixture into a methane-air mixture in an internal combustion engine cylinder. The domain is discretized using 60.2 Billion cells, with a total of 2.4 Trillion degrees of freedom. (a) Co-kurtosis metric M for AMR blocks in sixth level of refinement, colored by value (from blue to red); (b) Detailed view of AMR blocks in the highest level of refinement (level 6) colored by anomaly metric M (from blue to red). Zoomed-in view (green circle) shows the AMR blocks in more detail; (c) Same detailed view as in panel b, but only for blocks with anomaly metric $M > 0.65$. *Note: Figures were produced a posteriori using Python and Paraview.*

the future of high-energy physics colliders Albert et al. (2021). WarpX is built on top of the AMReX library and is an example of the value of integrating the co-design AMReX suite into ALPINE infrastructures. In this case, the integration of Ascent and AMReX created an easy path to publish WarpX simulation mesh data to Ascent. WarpX was integrated with Ascent and tested on OLCF's supercomputer, Frontier, at varying scales. Figures 10 and 11 are in situ renderings from Ascent of a staged laser-wakefield accelerator in a boosted reference frame. In these images an electron beam (orange-green) is accelerated to the right through multiple stages to high energies. And in the plasma stages (gray), the strong traversal focusing fields are shown in red-blue. To create these images, Ascent utilized VTK-m Moreland et al. (2016) to first transform the data via scaling, isosurfacing, and clipping, before rendering the final images. Ascent also utilized RAJA Beckingsale et al. (2019) to combine multiple electron fields into one to allow for volume rendering.

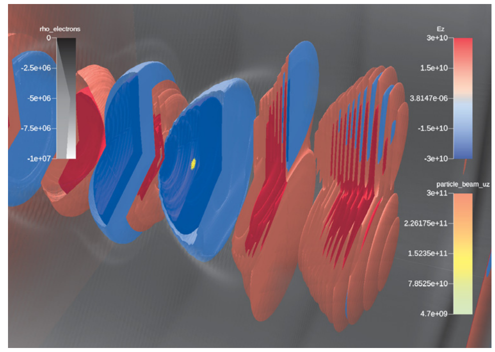

Figure 10. Visualization of a staged laser-plasma accelerator simulation. Shown is the strong traversal focusing fields (red-blue) in the first plasma stage (gray) and injected into this structure is an electron beam (orange-green) that is accelerated to the right to high energies. This in situ rendering of a later time step of the WarpX simulation executed on 552 GPUs across 69 nodes of Frontier.

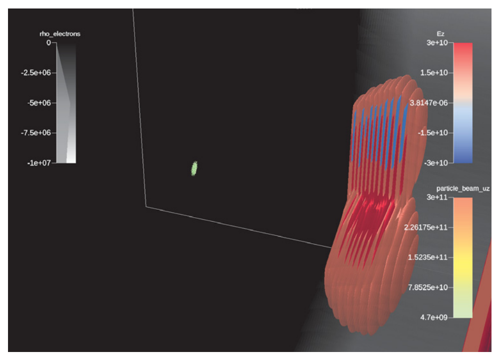

Figure 11. This in situ rendering of an early time step of the WarpX simulation was executed on 4,416 GPUs across 552 nodes of Frontier.

MFIX-Exa In situ Visualization with Catalyst

MFIX-Exa is a multiphase flow code developed to utilize the massive scale parallelism offered by the modern supercomputers while being performant and portable Musser et al. (2022). It relies on the AMReX Zhang et al. (2019) library, which provides a collection of efficient iterators, linear solvers, and communication routines on structured data and particles. To address the data management challenges posed by massive parallelism, MFIX-Exa added support for in situ visualization and analysis using Catalyst. This integration benefited both the products mutually as Catalyst and its ParaView backend unlocked access to an almost exhaustive suite of visual analytics for MFIX, and to support the needs of MFIX-Exa, Catalyst had to develop new protocols to handle AMReX data. This integration was tested on varying scales, and Catalyst was able to run on up to 649 nodes while using 5187 GPUs on Frontier. Figure

12 showcases the type of output images generated using this integration.

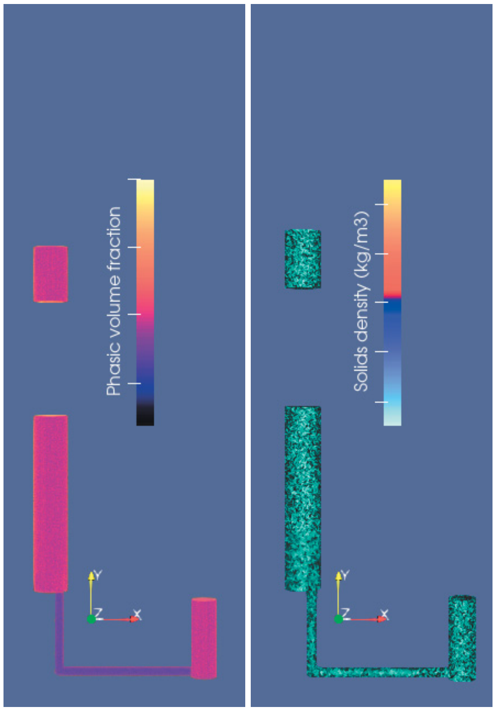

Figure 12. The MFIX-Exa team ran several intermediate-sized simulations using the Catalyst integration to generate in situ graphics. The above visualizations were demonstrated utilizing 30 nodes and 239 GPUs. This figure shows (left) rendering of the mesh outline of the reactor and (right) rendering of particles of the fluid phase volume fraction within the chemical looping reactor during the initial condition. Darker colors represent areas higher in solids concentration, whereas brighter colors are areas with few particles.

Conclusions

Exascale supercomputing architectures challenged the traditional post hoc visualization and analysis approaches because it is difficult to save simulation outputs at the rate they are generated. In addition, GPU accelerators required new algorithm and infrastructure implementations. The ALPINE project met these challenges by offering in situ algorithms and infrastructures. ALPINE infrastructures and algorithms are available to the community and can be found at the following sites:

- ParaView: https://www.paraview.org/
- ParaView GitLab: https://gitlab.kitware.com/paraview/paraview
- Catalyst Documentation: https://catalyst-in-situ.readthedocs.io/en/latest/
- VisIt: https://visit-dav.github.io/visit-website/
- Ascent GitHub: https://github.com/Alpine-DAV/ascent
- Ascent Documentation: https://ascent.readthedocs.io/en/latest/
- Algorithms: https://github.com/Alpine-DAV/algorithms/tree/master

Author Biographies

James Ahrens is a senior scientist at Los Alamos National Laboratory and currently serves as the director for LANL's Information Science and Technology Institute. His primary research interests are visualization, computer graphics, data science and high performance computing. Dr. Ahrens is author of over 140 peer reviewed papers that have been cited more than 8500 times. He is the founder/design lead of ParaView, an open-source visualization tool designed to handle extremely large data. ParaView is broadly used for scientific visualization, it has been downloaded close to three million times, and is in use at supercomputing and scientific centers worldwide.

Marco Arienti is a staff scientist at Sandia National Laboratories, CA, where he has developed high-fidelity models and simulations for aero- and automotive propulsion, focusing for the last 20 years on coupling shock-dynamics and combustion with fuel sprays. He has contributed to the development of the multi-physics package of the Pele codes for the Exascale Computing Project.

Utkarsh Ayachit is Senior Development Technology Engineer at NVIDIA. He is one of the core developers of ParaView and Catalyst. His interests include developing applications and toolkits for scientific computing workflows.

Janine Bennett is an R&D Science and Engineering Manager in the Quantum Information Sciences group at Sandia National Laboratories, California. She received her B.S. and Ph.D. in Computer Science from the University of California, Davis.

Roba Binyahib is a Graphics Software Engineer at Intel. Her work is focused on enabling large-scale rendering in the oneAPI rendering toolkit. She received her Ph.D. in Computer Science at the University of Oregon in the area of scientific visualization and high-performance computing. Her research interests include flow visualization, computer graphics, and in situ visualization.

Ayan Biswas is a scientist in the Information Sciences group at Los Alamos National Laboratory. He was the co-PI of *Insitu Inference*, a large-scale lab-directed R&D investment (2019–2022) and led the sampling efforts under Exascale Computing Project's (ECP) ALPINE project. Currently, he is the co-PI for the Advanced Simulation and Computing (ASC) program's LLMs for Code Translation project. He is also the deputy thrust lead for the foundation model thrust of the ArtIMis project, a large-scale LANL director's initiative investment to build Foundation models from scientific data. He is an expert in big data analytics, machine learning, visualization, and HPC systems.

Peer-Timo Bremer Peer-Timo holds a shared appointment at Lawrence Livermore National Laboratory's (LLNL) Center for Applied Scientific Computing (CASC), focusing on large-scale data analysis and visualization, and at the University of Utah, serving as Associate Director for Research of the Center for Extreme Data Management Analysis and Visualization (CEDMAV). His research interests include large-scale machine learning, data analysis, visualization, medical image analysis, topology, volume modeling, and virtual reality. Bremer joined LLNL in 2006. Prior to that, he was a post-doctoral research associate at the University of Illinois, Urbana-Champaign. He received a Ph.D. in computer science in 2004 from the University of California, Davis, and an M.S. and B.S. in mathematics and computer science from the Leipniz University, Hannover, Germany.

Eric Brugger has over 25 years of experience developing and using scientific visualization and analysis software. He is recently retired from Lawrence Livermore National Laboratory where he was the VisIt project leader and one of the original developers of the software. He received an R&D 100 award in 2005 as part of the development team of VisIt.

Roxana Bujack is a staff scientist in the Data Science at Scale Team at Los Alamos National Laboratory. She graduated in both mathematics and computer science and received her Ph.D. at Leipzig University. Her research interests include visualization, pattern detection, data science, HPC, moment invariants, and color theory.

Hamish Carr received his Ph.D. in Computer Science from the University of British Columbia in 2004, and is now a Professor at the University of Leeds. His areas of interest include computer graphics, scientific visualisation and computational topology. His work ranges from the foundational mathematics of visualisation through rendering problems and user interfaces to algorithmic development, topological analysis and domain-specific applications. Most recently, he has been working on scaling visualisation tools, and in particular topological tools, to modern hybrid exascale architectures.

Jieyang Chenis an Assistant Professor in the Department of Computer Science at the University of Alabama at Birmingham. He received his master and Ph.D. degrees in Computer Science from University of California, Riverside in 2014 and 2019. His research interests include high performance computing, parallel and distributed systems, and big data analytics.

Hank Childs is a professor of computer science at the University of Oregon. He received his Ph.D. from the University of California, Davis in 2006. Childs' research focuses on scientific visualization, high performance computing, and the intersection of the two.

Soumya Dutta is an Assistant Professor in the Department of Computer Science and Engineering (CSE) at the Indian Institute of Technology Kanpur (IIT Kanpur). Dutta obtained his Ph.D. in Computer Science and Engineering from the Ohio State University in May 2018. After his doctoral studies, Dutta worked as a postdoctoral researcher in the Applied Computer Sciences group (CCS-7) of Los Alamos National Laboratory (LANL) from June 2018 - July 2019. Then, he was promoted to a staff scientist position in the Information Sciences group (CCS-3) at LANL, where he worked from 2019–2022. His research interests are Machine Learning, Visual Computing, Uncertainty Visualization, xAI, In situ analysis, and HPC.

Abdelilah Essiari is a Computer Systems Engineer at Lawrence Berkeley National Laboratory (LBNL) with over 20 years of experience spanning distributed systems, security services, big data and software architecture. Essiari first joined LBNL in the late nineties, rejoining LBNL in 2020 after a stint at Cisco Systems as a Technical Lead and Gemini data as a Principal Engineer.

Berk Geveci leads the Scientific Computing team at Kitware Inc. His research interests include data analysis and visualization at scale, in situ analysis, and flow visualization. He has made significant contributions to the development of the Visualization Toolkit, ParaView and Catalyst.

Cyrus Harrison is a Computer Scientist and Deputy Division Leader in Lawrence Livermore National Laboratory's Computing Directorate. He develops data management, analysis, and visualization tools that support HPC multi-physics simulations. He is the software architect of the VisIt open source visualization tool and leads major aspects of the technical direction of the project.

Subhashis Hazarika is a Principal Researcher at Fujitsu Research of America (FRA). Subhashis obtained his Ph.D. in Computer Science and Engineering from the Ohio State University in 2019. After his doctoral studies, Subhashis worked as a postdoctoral researcher at the Applied Computer Sciences group of Los Alamos National Laboratory (LANL) from 2020–2021. He then worked as an Advanced Computer Scientist at Stanford Research Institute (SRI) from 2021–2024. Subhashis's current research interests lie at the intersection of machine learning, scientific visualization and climate AI.

Megan Hickman Fulp is a lecturer of computer science at Coastal Carolina University. She received her Masters in 2021 in Computer Engineering from Clemson University. Megan's research interests are focused on high performance computing, in situ scientific visualization, and big data reduction.

Petar Hristov is a postdoctoral researcher at the scientific visualization group at Linkoping University in Sweden. His research interests include topological analysis and scientific visualisation. This involves designing algorithms for topological data structures and working with scientists to analyze their data using topological methods.

Xuan Huang is a PhD student at the Scientific Computing and Imaging Institute at the University of Utah, under the supervision of Valerio Pascucci. Her research interests focus on scalable scientific data visualization systems and distributed computing of large-scale data.

Joseph Insley is the Team Lead for Visualization and Data Analysis at Argonne National Laboratory's Leadership Computing Facility, and Associate Professor in the School of Art & Design at Northern Illinois University. His research interests include scientific data analysis and visualization at extreme scale on current and future architectures. He collaborates with researchers from a wide range of science domains to provide insight into their data, and help communicate their findings to a broad audience.

Yuya Kawakami is a Ph.D. student in Computer Science at University of California, Davis. He completed his M.S. in Computer Science at the University of Oregon. His research interests include scientific visualization, information visualization and data storytelling.

Chloe Keilers is a staff scientist in the High Performance Computing Design group at Los Alamos National Laboratory. She has an M.S. in Human Computer Interaction from Rochester Institute of Technology and a B.S. in Computer Science from Stanford University.

James Kress is a Staff Scientist in the Visualization Core Laboratory at KAUST. His research and interests are focused on in situ scientific visualization, HPC, and the intersection of the two. James received his Ph.D. in 2020 and his Masters in 2017 in Computer Science both from the University of Oregon. He received his BS in 2013 from Boise State University in Computer Science, with a minor in Political Science. Prior to his time at KAUST, James spent 10 years at ORNL working on HPC, software development, and in situ visualization.

Matthew Larsen is a scientific software developer at Luminary Cloud specializing in scientific visualization. Larsen received an M.S. and Ph.D. in computer science from the University of Oregon. His research interests include computer graphics, scientific visualization, and HPC.

Dan Lipsa is a Staff R&D Engineer at Kitware. He earned his MS degree in computer science in 1998 and the Ph.D. degree in visualization from Swansea University in Wales, UK in 2013. Dan is a developer of the Visualization Toolkit (VTK), ParaView, and Catalyst.

Meghanto Majumder is a Ph.D. student in the Computing and Data Understanding at eXtreme Scale (CDUX) Lab in the Computer and Information Science Department at the University of Oregon. Majumder received B.Tech in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology.

Nicole Marsaglia is a Computer Scientist in Lawrence Livermore National Laboratory's Computing Directorate. She is a software developer on Ascent, a flyweight in situ analysis and visualization library, for HPC multi-physics simulations. Dr. Marsaglia received her BS in Pure and Applied Mathematics in 2014 and received her PhD in Computer Science in 2021 from the University of Oregon. Her research interests include automatic visualization, distributed algorithms, and in situ analysis.

Victor Mateevitsi is an Assistant Computer Scientist at the Argonne National Laboratory, where his research interests include large-scale visualizations, augmented and virtual reality technologies, and novel interaction techniques. Victor obtained his Ph.D. in Computer Science from the Electronic Visualization Laboratory at the University of Illinois at Chicago.

Valerio Pascucci is the founding Director of the Center for Extreme Data Management Analysis and Visualization (CEDMAV) of the University of Utah. Pascucci is also a Faculty of the Scientific Computing and Imaging Institute, a Professor of the School of Computing, University of Utah, and a Laboratory Fellow, of PNNL. Before joining the University of Utah, Pascucci was the Data Analysis Group Leader of the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory, and Adjunct Professor of Computer Science at the University of California Davis. Pascucci's research interests include Big Data management and analytics, progressive multi-resolution techniques in scientific visualization, discrete topology, geometric compression, computer graphics, computational geometry, geometric programming, and solid modeling. Pascucci is the coauthor of more than one hundred refereed journal and conference papers and has been an Associate Editor of the IEEE Transactions on Visualization and Computer Graphics.

John Patchett is in the Computer, Computational & Statistical Sciences division at Los Alamos National Laboratory (LANL). He is currently the Program Manager for the Advanced Simulation and Computing's Computational Systems & Software Environment program. Patchett received the B.A. degree, in 1995 and has worked at LANL since 1997. He received the M.S. degree in Computer Science from the University of New Mexico in 2011, and the Ph.D. degree from TU Kaiserslautern, Kaiserslautern, Germany, in 2017. His research interests include large-scale, parallel distributed memory, scientific data visualization and analysis for developing effective workflows for simulation data interpretation and understanding.

Saumil Patel is a scientist in Computational Sciences at Argonne National Laboratory. With a background in Computational Fluid Dynamics (CFD), his research interests include CFD, meshing techniques, high-order numerical methods, and high-performance computing. He received his Ph.D. in mechanical engineering from the Grove School of Engineering at the City College of New York (CCNY).

Steve Petruzza is an Assistant Professor at Utah State University. He received his M.S. and B.E. in computer engineering and his Ph.D. in computer science from the University of Rome "Tor Vergata" (Italy). His research activity started in the defense industry working on network centric multi-sensor data analysis and general purpose computing on GPUs and continued in the area of large scale scientific data visualization and high performance computing as a PostDoctoral fellow and later research associate at the Scientific Computing and Imaging Institute at the University of Utah. Steve's research interests are on large scale data movement, topological analysis, dynamic runtime systems and in situ workflows.

David Pugmire is a distinguished staff scientist at Oak Ridge National Laboratory. His research interests include methods for scientific visualization on high-performance computers. Pugmire has a Ph.D. in computer science from the University of Utah.

Silvio Rizzi is a Computer Scientist at Argonne National Laboratory. His research interests include High Performance Computing, Scientific Visualization, Virtual and Augmented Reality, and Display Technologies. Dr. Rizzi collaborates with several science teams and helps them understand the results of large-scale supercomputer simulations using scientific methods for data visualization and analysis.

David H. Rogers received the degree in architecture from Princeton University, the master's degree in computer science from the University of New Mexico, and the MFA degree in Writing for Children and Young Adults from the Vermont College of Fine Arts. His research interests include large scientific data visualization, image-based data analysis, and high-dimensional genomics visualization.

Oliver Rübel is a Staff Scientist in the Scientific Data Division at Lawrence Berkeley National Laboratory. He earned his MS degree in computer science in 2006 and the PhD degree in computer science in 2009 from the University of Kaiserslautern, Germany. His research interests include computational bioscience, high-performance data analysis and visualization, scientific data management, and machine learning.

Jorge Salinas obtained his Ph.D. in a joint degree program between the University of Florida and Balseiro Institute (Argentina). He is currently a postdoctoral appointee at Sandia National Laboratories. His research fields have spanned from high-fidelity submarine flow simulations and sediment transport, pathogen-laden droplet dispersal, anomaly detection in combustion and biometric data, and thermo-chemical nonequilibrium in extreme conditions.

Sudhanshu Sane is a software engineer and researcher focusing on visualization and HPC. Most recently, he was an engineer at Luminary Cloud working as a full stack developer building a cloud-based visualization service and the lead developer for flow visualization capabilities. He obtained his Ph.D. in Computer Science at the University of Oregon and was a postdoctoral research fellow in the Scientific Computing and Imaging Institute at the University of Utah. Prior to that he received an M.S. in CS from the University of Oregon and a B.Eng. in Information Technology from the University of Pune. His research interests include time-varying flow visualization, uncertainty and multi-field visualization, machine learning, and in situ analysis systems.

Sergei Shudler is a senior software engineer at e.solutions GmbH, where he works on a visualization system for an automotive infotainment platform. Previously, he was a performance engineer at Intel working on the SPECworkstation benchmarking suite. He obtained his doctorate degree at the Technical University of Darmstadt, Germany and completed his postdoc at Lawrence

Livermore National Laboratory. His research interests cover visualization and computer graphics, HPC, and performance analysis tools.

Alexandra Stewart has an M.S. from Georgia Tech in Computational Science and Engineering. She received her B.S. in Physics from the Massachusetts Institute of Technology. Her research interests include computational methods relevant to physics research in a variety of fields, including Machine Learning Theory, Scientific Computing, Probability Theory, High Performance Computing, and mesh and particle-based Simulations.

Karen Tsai has been a staff scientist in the Weapons Applications Research to Production (WARP) team at the Applied Computer Science group (CCS-7) of Los Alamos National Laboratory since graduation. She obtained her Master of Science in Computational Science, Engineering, and Mathematics from the University of Texas at Austin in May 2017. Her current work focuses on modern software engineering, integration of new technologies, and support of core capabilities in production applications to improve the stability, reproducibility, performance, scalability, portability, and agility of mission applications in production environments.

Terece L. Turton is a staff scientist in the Information Sciences group at Los Alamos National Laboratory. She received the B.S. degree in Physics from Carnegie Mellon University and the M.S. and Ph.D. degrees in Physics from the University of Michigan. Her current research interests include perceptual user evaluation and workflow development in scientific visualization.

Will Usher is a scientific visualization engineer at Luminary Cloud, where he works on a mix of challenging problems in computer graphics, parallel computing, and scientific visualization. Previously, he was a ray tracing software engineer at Intel working on CPU/GPU ray tracing, distributed rendering, and scientific visualization in OSPRay. He obtained his Ph.D. in Computer Science at the University of Utah, and prior to that a B.S in Physics at the University of California, Riverside. His research interests cover a range of areas in scientific visualization and computer graphics, across a spectrum of systems from HPC clusters to the web browser.

Zhe Wang is a Postdoctoral researcher in the Computer Science and Mathematics Division, Data and Artificial Intelligence Systems Section, Visualization Group at Oak Ridge National Laboratory. His work and research interests include scientific data visualization, adaptive in-situ workflow, and data management on high performance computing systems.

Gunther H. Weber is a Staff Scientist in the Scientific Data Division at the Lawrence Berkeley National Laboratory. He received a M.Sc. degree in Computer Science in 1999 and a Ph.D. in Computer Science in 2003, both from the University of Kaiserslautern, Germany. His primary research interests are data visualization, parallel computing, topological data analysis, and machine learning.

Corey Wetterer-Nelson is a Senior R&D Engineer at Kitware Inc. He received his PhD in Mechanical Engineering from the University of Colorado Boulder, focused on finite element methods, computational fluid dynamics, and high performance computing workflows. His current research interests include in situ analysis and visualization workflows and AI/ML co-processing workflows. Wetterer-Nelson is one of the lead developers of the Catalyst in situ analysis platform.

Jonathan Woodring is a computer scientist at Los Alamos National Laboratory. He received his Ph.D. from The Ohio State University in 2009. Woodring currently researches and develops simulation setup and initialization tools, including mesh generation and model linking across physical regimes.

Abhishek Yenpure is a Senior R&D Engineer at Kitware, where he focuses on parallel computing and scientific visualization and contributes to Kitware's open source offerings like ParaView, VTK and VTK-m, and Catalyst. He obtained his Ph.D. in Computer Science from the University of Oregon. His research interests include flow visualization, parallel algorithms, distributed algorithms, and in situ analysis.

Acknowledgements

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. This research is released under LA-UR-23-34194. The ALPINE team would like to thank all of our ECP and other collaborators who shared their time, software, data, and simulation codes with us.

References

Aditya K, Kolla H, Kegelmeyer WP, Shead TM, Ling J and Davis WL (2019) Anomaly detection in scientific data using joint statistical moments. *Journal of Computational Physics* 387: 522–538. DOI:https://doi.org/10.1016/j.jcp.2019.03.003. URL https://www.sciencedirect.com/science/article/pii/S002199911930172X.

Ahrens J, Jourdain S, O'Leary P, Patchett J, Rogers D and Petersen M (2015) An image-based approach to extreme scale in situ visualization and analysis. *International Conference for High Performance Computing, Networking, Storage and Analysis, SC* 2015: 424–434. DOI:10.1109/SC.2014.40.

Ahrens JP, Geveci B and Law CC (2005) Paraview: An end-user tool for large-data visualization. In: *The Visualization Handbook*, volume 717. Elsevier München. ISBN 9780123875822, pp. 717–731.

Albert F, Couprie ME, Debus A, Downer MC, Faure J, Flacco A, Gizzi LA, Grismayer T, Huebl A, Joshi C, Labat M, Leemans WP, Maier AR, Mangles SPD, Mason P, Mathieu F, Muggli P, Nishiuchi M, Osterhoff J, Rajeev PP, Schramm U, Schreiber J, Thomas AGR, Vay JL, Vranic M and Zeil K (2021) 2020 roadmap on plasma accelerators. *New Journal of Physics* 23(3): 031101. DOI:10.1088/1367-2630/abcc62.

Alexander FJ, Ang J, Bilbrey JA, Balewski J, Casey T, Chard R, Choi J, Choudhury S, Debusschere B, DeGennaro AM, Dryden N, Ellis JA, Foster I, Cardona CG, Ghosh S, Harrington P, Huang Y, Jha S, Johnston T, Kagawa A, Kannan R, Kumar N, Liu Z, Maruyama N, Matsuoka S, McCarthy E, Mohd-Yusof J, Nugent P, Oyama Y, Proffen T, Pugmire D, Rajamanickam S, Ramakrishniah V, Schram M, Seal SK, Sivaraman G, Sweeney C, Tan L, Thakur R, Essen BV, Ward L, Welch P, Wolf M, Xantheas SS, Yager KG, Yoo S and Yoon BJ (2021) Co-design center for exascale machine learning technologies (exalearn). *The International Journal of High Performance Computing Applications* 35(6): 598–616. DOI: 10.1177/10943420211029302. URL https://doi.org/10.1177/10943420211029302.

Anderson R, Andrej J, Barker A, Bramwell J, Camier JS, Cerveny J, Dobrev V, Dudouit Y, Fisher A, Kolev T, Pazner W, Stowell M, Tomov V, Akkerman I, Dahm J, Medina D and Zampini S (2021) Mfem: A modular finite element methods library. *Computers & Mathematics with Applications* 81: 42–74. DOI:https://doi.org/10.1016/j.camwa.2020.06.009. URL https://www.sciencedirect.com/science/article/pii/S0898122120302583. Development and Application of Open-source Software for Problems with Numerical PDEs.

Ayachit U, Bauer AC, Boeckel B, Geveci B, Moreland K, O'Leary P and Osika T (2021) Catalyst Revised: Rethinking the ParaView in Situ Analysis and Visualization API. In: Jagode H, Anzt H, Ltaief H and Luszczek P (eds.) *High Performance Computing*. Cham: Springer International Publishing. ISBN 978-3-030-90539-2, pp. 484–494.

Beckingsale DA, Burmark J, Hornung R, Jones H, Killian W, Kunen AJ, Pearce O, Robinson P, Ryujin BS and Scogland TR (2019) Raja: Portable performance for large-scale scientific applications. In: *2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*. pp. 71–81. DOI:10.1109/P3HPC49587.2019.00012.

Biswas A, Dutta S, Lawrence E, Patchett J, Calhoun JC and Ahrens J (2021) Probabilistic data-driven sampling via multi-criteria importance analysis. *IEEE Transactions on Visualization and Computer Graphics* 27(12): 4439–4454. DOI:10.1109/TVCG.2020.3006426.

Biswas A, Dutta S, Turton TL and Ahrens J (2022) Sampling for scientific data analysis and reduction. In: Childs H, Bennett JC and Garth C (eds.) *In Situ Visualization for Computational Science*. Cham: Springer International Publishing, pp. 11–36.

Borghesi G, Krisman A, Lu T and Chen JH (2018) Direct numerical simulation of a temporally evolving air/n-dodecane jet at low-temperature diesel-relevant conditions. *Combustion and Flame* 195: 183–202. DOI:https://doi.org/10.1016/j.combustflame.2018.02.020. URL https://www.sciencedirect.com/science/article/pii/S0010218018300920. Special Commemorative Issue: Professor Chung King (Ed) Law 70th Birthday.

Bujack R and Hagen H (2017) Moment invariants for multi-dimensional data. In: *Modeling, Analysis, and Visualization of Anisotropy*. Springer, pp. 43–64.

Bujack R, Rogers DH and Ahrens J (2018) Reducing occlusion in cinema databases through feature-centric visualizations. In: *Leipzig Symposium on Visualization In Applications (LEVIA)*, volume 2.

Bujack R, Zhang X, Suk T and Rogers D (2022) Systematic generation of moment invariant bases for 2d and 3d tensor fields. *Pattern Recognition* 123: 108313.

Carr H, Snoeyink J and Axen U (2003) Computing contour trees in all dimensions. *Computational Geometry: Theory and Applications* 24(2): 75–94.

Carr HA, Rubel O, Weber GH and Ahrens JP (2022a) Optimization and augmentation for data parallel contour trees. *IEEE Transactions on Visualization & Computer Graphics* 28(10): 3471–3485. DOI:10.1109/TVCG.2021.3064385.

Carr HA, Rübel O and Weber GH (2022b) Distributed hierarchical contour trees. In: *2022 IEEE 12th Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 1–10. DOI:10.1109/LDAV57265.2022.9966394.

Carr HA, Weber GH, Sewell CM, Rübel O, Fasel P and Ahrens JP (2021) Scalable contour tree computation by data parallel peak pruning. *IEEE Transactions on Visualization and Computer Graphics* 27(4): 2437–2454. DOI:10.1109/TVCG.2019.2948616.

Childs H, Brugger E, Whitlock B, Meredith J, Ahern S, Pugmire D, Biagas K, Miller M, Harrison C, Weber GH, Krishnan H, Fogal T, Sanderson A, Garth C, Bethel EW, Camp D, Rübel O, Durant M, Favre JM and Navrátil P (2012) Visit: An end-user tool for visualizing and analyzing very large data. In: *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*. Chapman and Hall/CRC, pp. 357–372. DOI:10.1201/b12985.

DAV SDK (2023) Ecp data and visualization sdk. https://ecp-data-vis-sdk.github.io/. Accessed 1/15/24.

DevilRay (2019) Devil Ray – A Portably Performant Ray Tracer for High-Order Element Visualization. URL https://github.com/LLNL/devil_ray.

Di S and Cappello F (2016) Fast error-bounded lossy hpc data compression with sz. In: *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. pp. 730–739. DOI:10.1109/IPDPS.2016.11.

Dutta S, Lipsa D, Turton TL, Geveci B and Ahrens J (2022a) In situ analysis and visualization of extreme-scale particle simulations. In: Anzt H, Bienz A, Luszczek P and Baboulin a (eds.) *High Performance Computing. ISC High Performance 2022 International Workshops*. Cham: Springer International Publishing, pp. 283–294.

Dutta S, Turton T, Rogers D, Musser JM, Ahrens J and Almgren AS (2022b) In situ feature analysis for large-scale multiphase flow simulations. *Journal of Computational Science* 63: 101773. DOI:https://doi.org/10.1016/j.jocs.2022.101773. URL https://www.sciencedirect.com/science/article/pii/S1877750322001533.

Fedeli L, Huebl A, Boillod-Cerneux F, Clark T, Gott K, Hillairet C, Jaure S, Leblanc A, Lehe R, Myers A, Piechurski C, Sato M, Zaim N, Zhang W, Vay JL and Vincenti H (2022) Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-In-Cell Simulations on Exascale-Class Supercomputers. In: *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. pp. 1–12. DOI:10.1109/SC41404.2022.00008. *Winning Paper, 2022 ACM Gordon Bell Prize*.

Harrison C, Larsen M, Ryujin BS, Kunen A, Capps A and Privitera J (2022) Conduit: A successful strategy for describing and sharing data in situ. In: *2022 IEEE/ACM International Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1–6. DOI:10.1109/ISAV56555.2022.00006. URL https://doi.ieeecomputersociety.org/10.1109/ISAV56555.2022.00006.

Henry de Frahan MT, Esclapez L, Rood J, Wimer NT, Mullowney P, Perry BA, Owen L, Sitaraman H, Yellapantula S, Hassanaly M, Rahimi MJ, Martin MJ, Doronina OA, A SN, Rieth M, Ge W, Sankaran R, Almgren AS, Zhang W, Bell JB, Grout R, Day MS and Chen JH (2024) The pele simulation suite for reacting flows at exascale. *Proceedings of the 2024 SIAM Conference on Parallel Processing for Scientific Computing* : 13–25DOI:10.1137/1.9781611977967.2. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977967.2.

Heroux M, Shende S and Willenbring J (2023) The extreme-scale scientific software stack. https://ecp-data-vis-sdk.github.io/. Accessed 1/15/24.

Hristov P, Weber GH, Carr HA, Rübel O and Ahrens JP (2020) Data parallel hypersweeps for in situ topological analysis. In: *2020 IEEE 10th Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 12–21. DOI:10.1109/LDAV51489.2020.00008.

Larsen M, Ahrens J, Ayachit U, Brugger E, Childs H, Geveci B and Harrison C (2017) The alpine in situ infrastructure: Ascending from the ashes of strawman. In: *Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization*, ISAV'17. New York, NY, USA: Association for Computing Machinery. ISBN 9781450351393, p. 42–46. DOI:10.1145/3144769.3144778. URL https://doi.org/10.1145/3144769.3144778.

Larsen M, Brugger E, Childs H, Eliot J, Griffin K and Harrison C (2015) Strawman: A batch in situ visualization and analysis infrastructure for multi-physics simulation codes. In: *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV2015. New York, NY, USA: Association for Computing Machinery. ISBN 9781450340038, p. 30–35. DOI:10.1145/2828612.2828625. URL https://doi.org/10.1145/2828612.2828625.

Larsen M, Harrison C, Turton TL, Sane S, Brink S and Childs H (2021) Trigger happy: Assessing the viability of trigger-based in situ analysis. In: *2021 IEEE 11th Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 22–31. DOI:10.1109/LDAV53230.2021.00010.

Larsen M, Woods A, Marsaglia N, Biswas A, Dutta S, Harrison C and Childs H (2018) A flexible system for in situ triggers. In: *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV '18. New York, NY, USA: Association for Computing Machinery. ISBN 9781450365796, p. 1–6. DOI:10.1145/3281464.3281468. URL https://doi.org/10.1145/3281464.3281468.

Lawson M, Harrison C, Brugger E, Skinner A and Larsen M (2021) Automated in situ computational steering using ascent's capable yes-no machine. In: *ISAV'21: In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV'21. New York, NY, USA: Association for Computing Machinery. ISBN 9781450387156, p. 24–30. DOI:10.1145/3490138.3490144. URL https://doi.org/10.1145/3490138.3490144.

Lindstrom P (2014) Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics* 20(12): 2674–2683. DOI:10.1109/TVCG.2014.2346458.

Marsaglia N, Kawakami Y, Schwartz SD, Fields S and Childs H (2021) An entropy-based approach for identifying user-preferred camera positions. In: *2021 IEEE 11th Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 73–83. DOI:10.1109/LDAV53230.2021.00015.

Marsaglia N, Majumder M and Childs H (2022a) A trigger-based approach for optimizing camera placement over time. In: *2022 IEEE/ACM International Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV)*. pp. 14–19. DOI:10.1109/ISAV56555.2022.00008.

Marsaglia N, Mathai M, Fields S and Childs H (2022b) Automatic In Situ Camera Placement for Isosurfaces of Large-Scale Scientific Simulations. In: Bujack R, Tierny J and Sadlo F (eds.) *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association. ISBN 978-3-03868-175-5, pp. 49–59. DOI:10.2312/pgv.20221065.

Moreland K, Lo LT, Pugmire D, Bolstad M and Childs H (2024) Viskores: A visualization library for many-threaded devices. URL https://github.com/Viskores/viskores.

Moreland K, Sewell C, Usher W, Lo Lt, Meredith J, Pugmire D, Kress J, Schroots H, Ma KL, Childs H, Larsen M, Chen CM, Maynard R and Geveci B (2016) Vtk-m: Accelerating the visualization toolkit for massively threaded architectures. *IEEE Computer Graphics and Applications* 36(3): 48–58. DOI:10.1109/MCG.2016.48.

Morozov D and Peterka T (2016) Block-parallel data analysis with DIY2. In: *6th IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. pp. 29–36. DOI:10.1109/LDAV.2016.7874307.

Musser J, Almgren AS, Fullmer WD, Antepara O, Bell JB, Blaschke J, Gott K, Myers A, Porcu R, Rangarajan D, Rosso M, Zhang W and Syamlal M (2022) Mfix-exa: A path toward exascale cfd-dem simulations. *The International Journal of High Performance Computing Applications* 36(1): 40–58. DOI:10.1177/10943420211009293. URL https://doi.org/10.1177/10943420211009293.

Petruzza S, Treichler S, Pascucci V and Bremer PT (2018) Babelflow: An embedded domain specific language for parallel analysis and visualization. In: *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, pp. 463–473.

Sane S, Bujack R and Childs H (2018) Revisiting the evaluation of in situ lagrangian analysis. In: *EGPGV@ EuroVis*. pp. 63–67.

Sane S and Childs H (2022) Exploratory time-dependent flow visualization via in situ extracted lagrangian representations. In: Childs H, Bennett JC and Garth C (eds.) *In Situ Visualization for Computational Science*. Cham: Springer International Publishing, pp. 91–109.

Sane S, Johnson CR and Childs H (2021a) Investigating in situ reduction via lagrangian representations for cosmology and seismology applications. In: Paszynski M, Kranzlmüller D, Krzhizhanovskaya VV, Dongarra JJ and Sloot PMA (eds.) *Computational Science – ICCS 2021*. Cham: Springer International Publishing, pp. 436–450.

Sane S, Johnson CR and Childs H (2022) Demonstrating the viability of lagrangian in situ reduction on supercomputers. *Journal of Computational Science* 61: 101615. DOI:https://doi.org/10.1016/j.jocs.2022.101615. URL https://www.sciencedirect.com/science/article/pii/S1877750322000400.

Sane S, Yenpure A, Bujack R, Larsen M, Moreland K, Garth C, Johnson CR and Childs H (2021b) Scalable In Situ Computation of Lagrangian Representations via Local Flow Maps. In: Larsen M and Sadlo F (eds.) *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association. ISBN 978-3-03868-138-0, pp. 19–29. DOI:10.2312/pgv.20211040.

Shudler S and Bremer PT (2022) Streaming statistics. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Shudler S, Petruzza S, Pascucci V and Bremer PT (2021) Portable and composable flow graphs for in situ analytics. In: *2021 IEEE 11th Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE, pp. 63–72.

The HDF Group (2006) HDF5 website. https://www.hdfgroup.org/solutions/hdf5/. Accessed 1/15/24.

Tsai KC, Bujack R, Geveci B, Ayachit U and Ahrens J (2020) Approaches for In Situ Computation of Moments in a Data-Parallel Environment. In: Frey S, Huang J and Sadlo F (eds.) *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association. ISBN 978-3-03868-107-6, pp. 57–68. DOI:10.2312/pgv.20201075.

VTK-h (2019) VTK-h – A toolkit of scientific visualization algorithms for emerging processor architectures. URL https://github.com/Alpine-DAV/vtk-h.

Westfall PH (2014) Kurtosis as peakedness, 1905–2014. r.i.p. *The American Statistician* 68(3): 191–195. DOI:10.1080/00031305.2014.917055. URL https://doi.org/10.1080/00031305.2014.917055. PMID: 25678714.

Zhang W, Almgren A, Beckner V, Bell J, Blaschke J, Chan C, Day M, Friesen B, Gott K, Graves D, Katz M, Myers A, Nguyen T, Nonaka A, Rosso M, Williams S and Zingale M (2019) AMReX: A Framework for Block-Structured Adaptive Mesh Refinement. *Journal of Open Source Software* 4(37): 1370. DOI:10.21105/joss.01370.