Journal Title
XX(X):1–18
©The Author(s) 2019
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Priority Research Directions for In Situ Data Management: Enabling Scientific Discovery from Diverse Data Sources

Tom Peterka,[1] Deborah Bard,[2] Janine C. Bennett,[3] E. Wes Bethel,[4] Ron A. Oldfield,[5] Line Pouchard,[6] Christine Sweeney,[7] and Matthew Wolf[8]

Abstract

In January 2019, the U.S. Department of Energy, Office of Science program in Advanced Scientific Computing Research, convened a workshop to identify priority research directions for in situ data management (ISDM). A fundamental finding of this workshop is that the methodologies used to manage data among a variety of tasks in situ can be used to facilitate scientific discovery from many different data sources—simulation, experiment, and sensors, for example—and that being able to do so at numerous computing scales will benefit real-time decision-making, design optimization, and data-driven scientific discovery. This article describes six priority research directions identified by the workshop, that highlight the components and capabilities needed for ISDM to be successful for a wide variety of applications—making ISDM capabilities more pervasive, controllable, composable, and transparent, with a focus on greater coordination with the software stack and a diversity of fundamentally new data algorithms.

Keywords

In situ data management

Introduction

Scientific computing will increasingly incorporate a number of different tasks that need to be managed. For example, SC18 Supercomputing Conference (2018) featured in situ analytics, big data, workflows, data-intensive science, machine learning, deep learning, and graph analytics—applications unheard of in a high-performance computing (HPC) conference just a few years ago. Perhaps most surprising, more than half of the 2018 Gordon Bell finalists featured some form of artificial intelligence, deep learning, graph analysis, or experimental data analysis in conjunction with or instead of a single computational model that solves a system of differential equations.

The U.S Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) convened a workshop on in situ data management (ISDM) on January 28–29, 2019 Peterka et al. (2019). This article provides background information on ISDM and information about the purpose of workshop and summarizes the outcomes and findings of the workshop.

In this article as in the workshop, **we define ISDM as the practices, capabilities, and procedures to control the organization of data and enable the coordination and communication among heterogeneous tasks, executing simultaneously in an HPC system, cooperating toward a common objective**. This workshop considered in situ data management, in addition to its traditional roles of accelerating simulation I/O and visualizing simulation results, to more broadly support future scientific computing needs (Figure 1). The workshop identified priority research directions (PRDs) for ISDM to support current and future HPC scientific workloads, which include the convergence of simulation, data analysis, and artificial intelligence, requiring machine learning, data manipulation, creation of data products, assimilation of experimental and observational data, analysis across ensemble members, and, eventually, the incorporation of tasks on non-von Neumann architecture.

The I/O bottleneck is one driver of in situ analysis. Disparity in data movement latency, bandwidth, and energy consumption compared with the rate of floating-point operations has led to a renewed interest in ISDM. To put the imbalance between computing and data management in perspective, the rate of data that can be computed on the Summit Oak Ridge Leadership Computing Facility (2019) supercomputer (assuming 1 byte generated per clock cycle) is five orders of magnitude greater than the bandwidth of its parallel file system.

Current approaches to manage this bottleneck focus on data analysis, visualization, and the production of data products in situ. The resulting data products are often orders of magnitude smaller than the full state data,

[1]Argonne National Laboratory, Lemont, IL, USA
[2]National Energy Research Scientific Computing Center, Berkeley, CA, USA
[3]Sandia National Laboratories, Livermore, CA, USA
[4]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[5]Sandia National Laboratories, Albuquerque, NM, USA
[6]Brookhaven National Laboratory, Upton, NY, USA
[7]Los Alamos National Laboratory, Los Alamos, NM, USA
[8]Oak Ridge National Laboratory, Oak Ridge, TN, USA

Corresponding author:
Tom Peterka, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Lemont, IL 60439, USA
Email: tpeterka@mcs.anl.gov

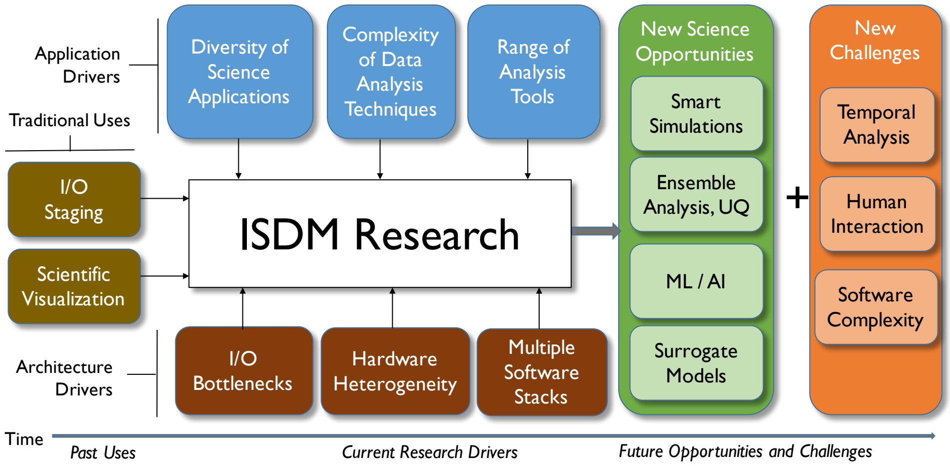

Figure 1. Changing role of ISDM over time and the motivations, opportunities, and potential challenges for a renewed research effort in this area.

thereby eliminating some of the negative impacts of the I/O bottleneck and saving storage space. In situ analyses can also lead to better science. While the infrequency of data outputs limits the fidelity of post hoc analysis, in situ analysis can have much higher fidelity because analysis tasks have access to simulation data directly and are not throttled by the I/O. The in situ paradigm, however, also complicates some operations. For example, human interaction, exploratory investigation, and temporal analysis are easier to conduct post hoc. In situ methods also add complexity to the workflow because of the larger number of interconnected, concurrent tasks that need to be managed.

A survey of past and present in situ methods and tools Bauer et al. (2016) demonstrates how reusable in situ software evolved separately from the storage and visualization communities. Storage solutions originally were used for staging a simulation's state for checkpointing, restarting, or saving outputs for later post hoc analysis. Even though such tools have expanded their applications beyond I/O staging, their I/O style of interface and data model remain. Meanwhile, the scientific visualization community developed in situ equivalents of their post hoc tools. Coming at the in situ problem from a visualization direction, these tools feature the VTK data model and scripts for connecting and executing pipelines of VTK filters.

A motivation for this workshop is that ISDM capabilities could be expanded and leveraged for a broader range of current and future HPC applications beyond I/O staging and scientific visualization. In addition to helping meet the challenges of extreme-scale simulation data, ISDM technologies can facilitate applications that merge simulation and data analysis, simulation and machine learning, or the processing and analysis of experimental data. This workshop identified a diversity of future workloads, listed below, for which ISDM has the potential to enable new capabilities.

- Smart simulations featuring online feedback, computational steering, multiphysics, and/or surrogate modeling
- Ensemble analysis of stochastic or rare events, uncertainty studies, or model calibration
- High-fidelity, highly scalable data analysis and visualization for debugging, diagnostics, and high temporal and spatial resolution
- Workflows featuring the convergence of big data and HPC software and tools: for example, graph analytics, database storage, and streaming
- Use of machine learning and deep learning alongside simulations or experiments for data-driven analysis methods
- Real-time experimental and observational data analysis and assimilation of streaming, potentially noisy, and time-critical data

As a concrete example, the workflow in Figure 2 illustrates some of the above elements: an ensemble of numerous instances of molecular dynamics simulations is launched and analyzed, searching for the signature of the stochastic process of nucleation as a material cools and crystallizes Yildiz et al. (2019). Crystal structures are detected in situ, and only features of interest are saved to storage. Instead of one large simulation, many smaller instances are launched dynamically until a rare event is detected—a pattern that has widespread applicability to other domains such as protein folding, self-assembled structures, and genetic algorithms.

The workflow of Figure 2 is represented as a directed graph: nodes are tasks, and edges represent data flow between two tasks. In the past, in situ usage was often limited to a single analysis or visualization task coupled to one simulation. Modern workflows have multiple tasks, can contain cycles with feedback, and can vary dynamically over time. Dynamic task graphs, for example for machine learning and artificial intelligence problems, can contain thousands of tasks because of the large amount of training needed to learn complex nonlinear scientific behavior.

Today, new tools are being developed for generic data producer/consumer tasks with the potential to manage a general graph of tasks communicating custom data types. Lacking, however, is a common vision for core capabilities to be delivered to users, as well as sufficient attention to making these tools interoperable. To more broadly support scientific computing needs, the workshop provided a forum to address

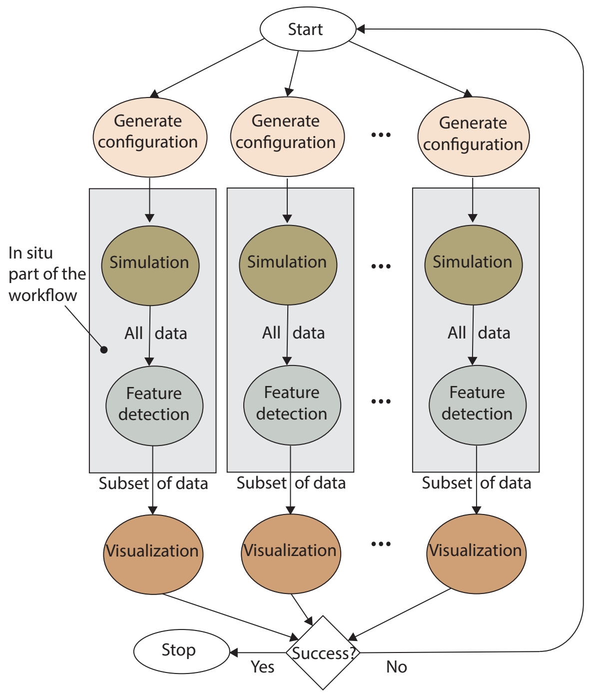

Figure 2. Workflow of dynamic ensemble of simulations and in situ detection of stochastic events.

generic in situ data management capabilities, for example for machine learning, automated spawning of ensemble runs, automated triggering and production of data products, and tasks run on non-von Neumann architectures. Workshop participants also had opportunities to discuss provenance and uncertainty as data are managed across tasks, as well as ways to facilitate workflows across multiple data and computing resources through interfaces between distributed and in situ workflow systems.

The outcomes of the workshop are distilled into six priority research directions, illustrated in Figure 3. The PRDs highlight the components and capabilities needed for ISDM to be successful for the wide variety of applications discussed: making ISDM capabilities more pervasive, controllable, composable, and transparent, with a focus on greater coordination with the software stack and a diversity of fundamentally new data algorithms.

Background

Scientific Workflows

A scientific workflow is a set of tasks or programs that cooperate in terms of scheduling and communication as part of a larger scientific campaign. Workflows are often characterized in two types—distributed workflows or in situ workflows—although in practice both types are combined in hybrid ways within science campaigns. Following the definition by Deelman et al. Deelman et al. (2017), an in situ workflow's tasks are tightly coupled, execute in one centralized computing system or facility, and exchange information over the memory hierarchy and network of that system; whereas a distributed workflow is one whose tasks are more loosely coupled, exchange data through files or remote connections, and may execute on geographically distributed systems.

Workflow management systems (WMSs) for distributed computing were originally developed for grid environments. Representative examples of distributed WMSs are described in several surveys Yu and Buyya (2005); Deelman et al. (2009) and include FireWorks Jain et al. (2015), Parsl Babuji et al. (2019), Pegasus Deelman et al. (2015), and the Sandia Analysis Workbench Friedman-Hill et al. (2015).

In situ workflows, in contrast, launch all tasks concurrently in one HPC facility, and communication occurs over shared memory or through the interconnect of the machine. Examples include Alpine Larsen et al. (2017), ParaView Ayachit (2015); Ayachit et al. (2015), VisIt Childs et al. (2012), Adios Lofstead et al. (2008); Liu et al. (2014), SENSEI Ayachit et al. (2016), Decaf Dreher and Peterka (2017), and Damaris Dorier et al. (2016).

A dichotomy remains between in situ and distributed workflows despite efforts to integrate these communities Deelman et al. (2016). Even though heterogeneous hierarchical combinations of multiple workflow models are starting to appear in the research literature Yildiz et al. (2019), hybrid workflows in practice are limited to single use-cases. CyberShake, a seismic hazard model from the Southern California Earthquake Center Graves et al. (2011), combines high-performance and high-throughput computing. Synchrotron light sources combine HPC with detector hardware and in situ processing Khan et al. (2013). KBase, a systems biology knowledge base, contains in situ modeling and reconstruction while offloading other processing to cloud systems Benedict et al. (2014).

The definition of ISDM given in the introduction—controlling and organizing data to enable heterogeneous tasks to execute simultaneously in an HPC system—is closely related to in situ workflows. The distinction between workflows and data management is subtle, but important. Data management—the practices, capabilities, and procedures to control the organization of data—are the core building blocks that enable workflows. In the context of this workshop, in situ workflows are applications that use ISDM, and the aspects of ISDM being investigated are at a lower level of abstraction than the workflows and WMSs described above.

Computational Platforms

New hardware advances present exciting challenges and opportunities for ISDM. One example is the availability of large-scale nonvolatile random-access memory (NVRAM). Individual NVRAM abstraction layers (e.g., libhio Hjelm and Wright (2017), Data Elevator Dong et al. (2016), BurstMem Wang et al. (2014), DataWarp Henseler et al. (2016), Mochi Dorier et al. (2018); Jenkins et al. (2017); Carns et al. (2016)) exist, but community-wide standards have yet to evolve.

Another active area of research is the use of non-von Neumann and emerging hardware for artificial intelligence (AI), machine learning (ML) Schuman et al. (2017); James et al. (2017), and scientific computing Severa et al. (2016). These and other surveys Agarwal et al. (2016) point to the potential for significant energy savings, although the use of neuromorphic hardware for general-purpose scientific computing is still limited.

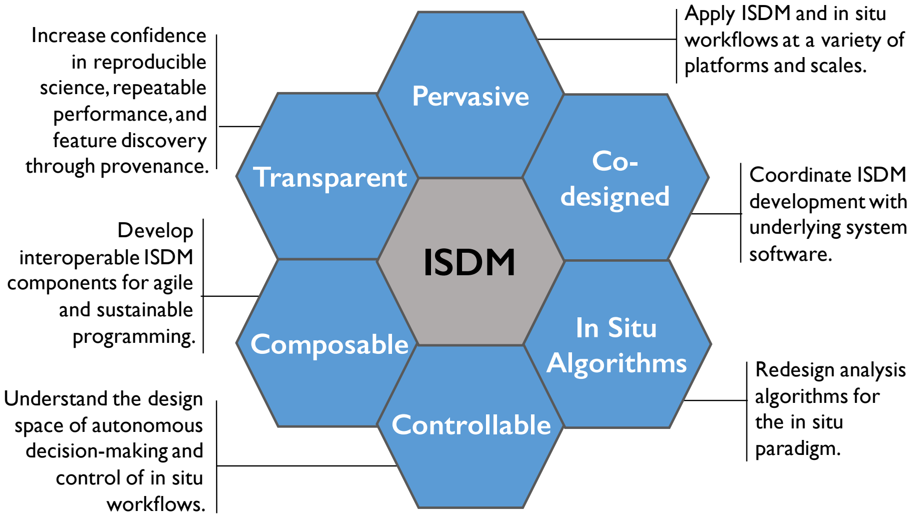

Figure 3. Priority research directions at a glance.

Operating system and run-time research funded by the Exascale Computing Project (ECP) and ASCR (Argo Perarnau et al. (2013), Hobbes Brightwell et al. (2013)) investigates system support for unconventional HPC programming models, support for multiple concurrent run-times, and advanced virtualization capabilities that could be leveraged to support desired ISDM capabilities. However, as Dreher et al. Dreher et al. (2017) show, HPC platforms still do not support all the capabilities needed for in situ workflows. Notably lacking in present-day HPC systems are dynamic process management, high-performance interjob communication, and intranode task isolation.

Early work has been done in integrating software stacks, primarily the Apache big data stack with the HPC stack. MapReduce models can use the Message Passing Interface (MPI) in various ways Caino-Lores et al. (2018); Malitsky et al. (2017); Bicer et al. (2017); Gao et al. (2017); Wang et al. (2015); Gittens et al. (2018). However, more work is needed to find a low-overhead integration that does not require changing the constituent programming models and allows existing tools to use the integrated software system.

Experimental and Observational Science

Programming models that support streaming of data, which is often required to connect experimental facilities with HPC centers, include Psana Damiani et al. (2016), ADARA Shipman et al. (2014), ICEE Choi et al. (2013), and the NSLS-II event model NSLS-II Data Acquisition and Management Group (2019). Software infrastructures addressing high-volume, high-throughput data streams from light source instruments include Xi-cam Pandolfi et al. (2018) and Nanosurveyor Daurer et al. (2017). The Bluesky library for experiment control and collection of scientific data and metadata Koerner et al. (2019); Allan et al. (2019) is being used at more than one light source user facility. The HEP community has identified the need for software architecture approaches for large-scale experiments, which have lifetimes that can span multiple decades Hildreth et al. (2018). Streaming of data from an instrument to HPC computing in the ATLAS project at CERN is being developed by Magini et al. Magini et al. (2018).

The National Energy Research Scientific Computing Center (NERSC), as part of its effort to support experimental and observational science, provides web-based APIs for accessing services National Energy Research Scientific Computing Center (2019) as well as real-time queues for time-critical workloads and networking infrastructure optimized for high-volume scientific data movement Dart et al. (2014).

Real-time Control

Modern workflows can incorporate many tasks coupled in a directed graph that can include multiple cycles; hence, control of ISDM systems is considerably more challenging and requires a greater degree of automation than in the past. Automatic identification of time steps of interest for deeper analysis include Larsen et al. (2018); Bennett et al. (2016); Salloum et al. (2015); Woodring et al. (2011); Dutta et al. (2018). Other works identify input parameter values based on simulation state Weber et al. (2007), enabling autonomous deployment of algorithms that historically have required a human in the loop.

The real-time control of a simulation is sometimes called computational steering. Early work includes coupling a simulation with data analysis and exploration tools for user-guided problem setup and monitoring Bethel et al. (1994). Another early example of computational steering is the SciRun Parker and Johnson (1995) system. In 2018, the Uintah code was coupled with VisIt through a dashboard Sanderson et al. (2018) offering steering capability.

Analysis Algorithms

In situ analysis algorithms may transform data into reduced representations or surrogate models in order to mitigate large data size, high dimensionality, or long computation times. Low-rank approximation Austin et al. (2016); statistical summarization Hazarika et al. (2018); Thompson et al. (2011); Biswas et al. (2018); Dutta et al.

(2017); Lawrence et al. (2017); Lohrmann et al. (2017); topological segmentation Morozov and Weber (2013, 2014); Gyulassy et al. (2012, 2019); Landge et al. (2014); wavelet transformation Li et al. (2017); Salloum et al. (2018); lossy compression Di and Cappello (2016); Lindstrom (2014); Brislawn et al. (2012); geometric modeling Peterka et al. (2018); Nashed et al. (2019); and feature detection Guo et al. (2017) may be used to generate reduced or surrogate models.

Information-theoretic methods Biswas et al. (2013); Wang and Shen (2011) can quantify the overall information content of a temporal or spatial interval, and changes in information entropy can indicate potential areas of further investigation. Machine learning methods Wozniak et al. (2018); Kurth et al. (2018); Joubert et al. (2018) can elucidate features that cannot be described by other methods.

User intent and constraints in an in situ system can be expressed through a file I/O interface such as the BP format in ADIOS Liu et al. (2014) as well as NetCDF Davis et al. (2017) and HDF5 Folk et al. (1999) formats; through generic data containers that are created through an API inside the user's tasks Dreher and Peterka (2016); or in data contracts Mommessin et al. (2017); Dorier et al. (2017); or through service-level agreements (SLAs) such as in CORBA OMG (2000). Self-describing and extensible interfaces, for example, Scientific XML Widener et al. (2002) and External Data Representation Srinivasan (1995), have been used for similar purposes.

Software Design

The increasing diversity of ISDM applications—observations, experiments, ensembles, edge computing, multifacility federated workflows, to name a few—require ISDM software that is available, stable, reusable, and maintainable. Today, however, ISDM lacks widespread tool and data model interoperability. There is consistency at the level of specific domain science communities, such as the ROOT programming and data model in HEP Antcheva et al. (2009) or the NetCDF file format in the climate community Davis et al. (2017).

Recent efforts in the DOE have begun to address software development and deployment standards for scientific computing. The IDEAS McInnes (2019); Balay et al. (2016) project was an early attempt to make math libraries interoperable by enforcing consistent guidelines for their development and delivery. These efforts have been extended through the ECP xSDK xSDK Project (2019); Bartlett et al. (2017). Delivery of ECP software is being done through the Spack Gamblin et al. (2015) package manager as well as through containers such as Shifter Kincade (2015) and Singularity Argonne Leadership Computing Facility (2019).

Provenance

Numerous standards and tools for provenance exist in communities other than the HPC and DOE. For instance, the W3C-Prov World Wide Web Consortium Working Group (2013) specifies standards for exchanging provenance information in heterogeneous environments. Standards for smart sensors such as the Open Geo Consortium Sensor Web Enablement Pouchard et al. (2009) encode the provenance of a sensor signal. Tools for replicability and reproducibility collect provenance, such as ReproZip Chirigati et al. (2013), and initiatives facilitate reproduction of studies based on provenance, such as the Open Science Framework Center for Open Science (2011).

Provenance software can alert developers of anomalous system behavior, but such tools require post hoc analysis, for instance, in the security domain Pasquier et al. (2018), and for Spark dataflows Interlandi et al. (2015). Provenance capture is I/O-intensive, although research Singh et al. (2016) has shown that supervised ML algorithms trained post hoc can alleviate the in situ I/O burden of provenance collection by performing intelligent triage.

Performance profiling tools for HPC exist, such as HPC Toolkit Tallent et al. (2008) and SONAR Lammel et al. (2016). TAU Shende and Malony (2006) and ScoreP Knpfer et al. (2012) extract performance profiles from applications, but these tools are not optimized for large-scale workflows, nor do they collect comprehensive provenance information required for detailed introspection and analysis. WOWMON Zhang et al. (2016) presented a solution for online monitoring and analytics of scientific workflows, but imposed several limitations and lacked generality in interfacing with workflow components. SOSflow Wood et al. (2016) adopts a general-purpose data model, runtime adaptivity, workflow configurability, and supports integration of analytics or visualization, although more research in configuration and deployment at scale remains. Chimbuko Pouchard et al. (2018) supports workflow-level performance analysis, but provenance is not analyzed in situ.

Provenance information also exists at the computing system or facility level. Monalytics Kutare et al. (2010) combines monitoring and analytics to rapidly detect and respond to complex events in large data centers. Tools such as Darshan Snyder et al. (2016) and TOKIO Lockwood et al. (2018) measure the performance of the I/O system and applications' interaction with it. Graphs have been explored Ames et al. (2013); Dai et al. (2014) for storage of such provenance metadata. HPC facilities use tools such as an automatic library-tracking database Fahey et al. (2010) and XALT Agrawal et al. (2014) to track which libraries are linked with which applications, information that can assist individual users as well as system administrators.

Workshop Report

This article is a summary of the detailed report of the ISDM workshop by Peterka et al. Peterka et al. (2019). The body of the report is divided into two main chapters: priority research directions (Chapter 2) and workshop topics (Chapter 3). Chapter 2 contains the six main outcomes—the priority research directions (PRDs)—of the workshop, which are also summarized in the following section of this article. Chapter 3 of the report contains the raw data that went into the findings of Chapter 2: a detailed account of the breakout discussions that transpired during the course of the workshop. The report is organized in highlights-to-details order: headlines appear early with supporting details following. A high-level overview can be gleaned from the executive summary; more context and detail follows in the introduction in Chapter 1; a comprehensive explanation of the PRDs follows in Chapter 2, with information from the discussion topics in Chapter 3.

There is not a one-to-one correspondence between the breakout discussion topics and background sections presented above to the PRDs. Rather, all of the PRDs cross-cut and combine multiple discussion topic areas. This is by design. The background and discussion topics were predetermined before the workshop: they were the inputs to the process. The outputs of the workshop are the synthesis of the discussions into a succinct set of research directions. The actual mapping of inputs to outputs, discussion topics to PRDs, is contained in the report and in Figure 4.

The wording of the PRDs is aspirational, describing desired capabilities that the community wants to acquire, the research needed to attain those capabilities, and the anticipated benefits of doing so.

Priority Research Directions for ISDM

The priority research directions described here highlight the components and capabilities needed for ISDM to be successful for a wide variety of applications.

Pervasive ISDM

Apply ISDM methodologies and in situ workflows on a variety of platforms and scales.

Key questions How can ISDM methodologies help meet the needs for real-time, high-velocity data applications at the edge and other non-HPC platforms? How can ISDM enable science at experimental and observational facilities? How do ISDM methodologies for traditional computational modeling compare with ISDM methodologies for experimental and observational facilities, including edge devices such as sensors and detectors? Can recognizing commonalities among such disparate use cases increase the adoption of ISDM among scientists across the DOE mission and bridge scientific communities?

Research opportunities A changing landscape of use cases is driving new applications of ISDM, which increasingly require ISDM near instruments or sensors for data analysis in near-real time. To what extent can similar ISDM approaches be applied both to HPC computing and to edge computing in experimental or observational facilities? The next generation of ISDM research will often require combining and coordinating workflows across multiple such computational platforms in order to answer fundamental science questions. Clearly needed, then, is the ability to execute the same ISDM tasks and workflows across a spectrum of computational platforms, spanning high-performance supercomputers to experimental detectors and even embedded devices. The workshop called this desired capability "pervasive ISDM." Pervasive ISDM requires streaming data over heterogeneous computing and networking scales and satisfying real-time demands of experimental instruments. Also essential are new algorithms and reduced data representations that are designed for low-power embedded processors with limited memory or bandwidth. Experimental data can be noisy, containing errors or missing data points; and assimilating real and simulated data requires managing disparate levels of data quality, in addition to disparities in scale, resolution, and data organization among the various data sources of the overall science workflow.

New Research Directions In order for the efficiencies and capabilities of ISDM to pervade across computational scales and platforms, a number of new research directions need to be pursued, which also motivate further exploration in the other five PRDs.

The increasing deployment of computing near experimental detectors, whether in traditional or specialized processors, complicates in situ workflows and necessitates the development of new ISDM tools to handle a mixture of computing resources and data signals. New research includes co-design among ISDM, system software, programming models, and hardware vendors.

Pervasive ISDM will require advances in data flow between real-time streams as opposed to bulk-synchronous checkpoints, as in traditional HPC. How to fuse multiple data channels such as instruments, human users, and other computing systems in ISDM applications, are open questions. Streaming APIs are needed to help connect data throughout the stages of the workflow, from instrument to computing cluster to HPC, and potentially back again. Data models developed for streaming data will need to incorporate both edge and HPC systems and the relevant metadata from multiple systems.

New algorithms also need to be developed for in situ analysis on edge devices, particularly for emerging hardware such as FPGAs or neuromorphic devices. Single-pass algorithms will generate reduced representations in a streaming regime, and research in automatic triggers and other decision-making capabilities will be important for control of experiments based on in situ analyses.

Provenance processing at the edge can detect real-time anomalies as data are generated. Unique provenance identification is needed to track data over a lifespan of movement across a variety of locations.

Composable programming and execution models that support streamed data for in situ processing will need to access both edge and HPC data, be scheduled across multiple platforms, and have standardized interfaces for both edge and HPC applications, as opposed to one-off solutions.

Potential benefits Considering the combined system of instruments and computing, scientists want to be able to place data analysis where it is required, based on timeliness and other constraints. Being able to deploy ISDM on a variety of platforms would enable this flexibility. Data reduction and analysis are key to dealing with high volume and velocity data, and performing some of these operations as early as possible will increase the efficiency of later processing stages. Pervasive ISDM would reduce human effort by reusing software tools, algorithms, and frameworks and would improve understanding of performance and science by applying consistent computing methods. The ability to deploy in situ approaches pervasively, across platforms and scales, would advance experimental, observational, and computational science.

Breakout Session Research Areas	PRDs					
	Pervasive ISDM	Co-designed ISDM	In Situ Algorithms	Controllable ISDM	Composable ISDM	Transparent ISDM
DATA MODELS						
Multimodal data science and ML	X		X	X		X
Intent expression and reuse	X	X	X	X	X	
COMPUTATIONAL PLATFORMS						
System software support		X				X
Heterogeneous hardware		X				
ANALYSIS ALGORITHMS						
Reduced representations	X		X			
Run-time control		X	X	X		
New platforms and outputs	X	X	X			X
PROVENANCE AND REPRODUCIBILITY						
Scalable, portable provenance capture	X	X				X
In situ provenance processing			X			X
Provenance for ML and reproducibility					X	X
PROGRAMMING AND EXECUTION MODELS						
Elastic, dynamic resources		X		X		
Scheduling and optimizing execution		X		X		X
Composable workflows	X	X	X		X	
Streaming	X		X			X
SOFTWARE ARCHITECTURE						
Use cases driving design	X				X	
Usability and sustainability					X	
Software tool interoperability					X	
User confidence					X	
Science facilities partnerships	X				X	

Figure 4. Mapping of workshop breakout session research areas to resulting priority research directions.

Co-designed ISDM

Coordinate the development of ISDM with the underlying system software so that it is part of the software stack.

Key questions What abstractions, assumptions, and dependencies on system services are needed by ISDM? What information must be exchanged between the ISDM tools and the rest of the computing software stack to maximize performance and efficiency? How can we ensure seamless communication between the ISDM software layer and other parts of the system software stack?

Research opportunities Understanding the interlayer dependencies so that ISDM becomes part of the software stack can facilitate connections between software layers, communicate semantic meaning, and realize efficient performance in HPC and other software stacks. Defining the dependencies between ISDM and the rest of the software stack can enable autonomous data management, efficient algorithmic performance, and verifiable science. Dependencies do not only extend *down* the software stack toward system software layers; just as important are *upward* connections toward application and workflow layers in higher levels of the software stack. There are also opportunities for connections *across* multiple software stacks, for example, HPC and big data.

New Research Directions Co-design among ISDM, system software, programming models, and computing vendors is needed to ensure being able to adapt and evolve with the various communities—and perhaps influence design to jointly meet needs of both computing architecture and ISDM research. Coordinated explorations should take place with vendors of emerging post-Moore and non–von Neumann hardware so that such hardware can benefit ISDM. Systems and hardware should be mapped to the needs of ISDM, exploiting the opportunity to align hardware to the needs of ISDM, and leading to improved utilization of computing platforms.

The storage/memory hierarchy is vital to ISDM. In the same way that the HPC community has been evolving performance-portability abstractions for CPUs and GPUs, there remains research potential for storage and memory abstractions in support of ISDM, particularly NVRAM. Abstractions for heterogeneous computing units are also needed for ISDM. Beyond performance-portability abstractions for CPU and GPU threads, programming models will also be needed for FPGAs, tensor processing units, neuromorphic chips, and eventually quantum cores.

New system software abstractions and APIs need to link ISDM with applications, workflow management systems, and HPC system software. For example, dynamic resource allocation and dynamic task management are needed in HPC systems. Scheduling support must be developed to provide real-time queues, reservations for experiments, and remote connections to other facilities. Security features need to allow an HPC system to accept remote connections (incoming and outgoing) to/from other facilities, and varying levels of security need to be available for individual users as well as members of science teams. High-speed network connectivity among multiple tasks in an ISDM workflow needs to be implemented and supported.

Specifications need to be developed detailing what information is exchanged between the system software and ISDM framework or tasks running within it. The specifications include not only scientific data format but also metadata conveying user intent. For example, the ISDM software ought to be able to tell the operating system and run-time (OS/R) about the desired quality of service or constraints imposed on resources, so that the OS/R can intelligently manage resources and data Conversely, the OS/R should relate profiling information to the ISDM software regarding system health and resource usage levels, so that the ISDM framework can adjust intent or quality levels.

Research should also be directed at support for multiple, concurrent software stacks. Rather than user-level add-ons and adaptors between different programming environments, full-system installation of multiple stacks in the same system is needed, as well as system-level integration or communication among the stacks. For example, the same underlying I/O system should support parallel file systems, key-value stores, and databases.

Potential benefits The co-design of new services, new platforms, and application workflows has the potential to revolutionize ISDM. Thinking of ISDM as an integral part of the computational platform has numerous advantages over developing ISDM software independently from the computing hardware, system software, and applications or workflows. Competition for computational and data resources among in situ tasks can be avoided. Significant power savings are realizable by using next-generation hardware such as neuromorphic chips or nonvolatile memory. Better communication across system software layers can save developer time, provide consistent interfaces to users, and improve the use of computing and data resources. Integration of other software stacks and frameworks (e.g., from industry or big data) can exploit economies of scale, because those tools often have many more developers contributing to them compared with HPC software. More complex workflows can be supported by co-design of new hardware and software systems informed by ISDM use cases.

In Situ Algorithms

Redesign data analysis algorithms for the in situ paradigm.

Key questions How should in situ algorithms be designed to make most of the available resources? What new classes of data transformations can profit from in situ data access in the presence of constraints imposed by other tasks? What algorithms are needed for multiscale, multimodal, and multiphysics in situ coupling of tasks and data?

Research opportunities The capability to access every datum of a computation or an experiment poses unique opportunities and challenges for algorithm design, for traditional visualization, analysis, and for ML and AI. The in situ environment for data processing and analysis differs substantially from the post hoc environment, requiring fundamentally new algorithms and approaches. Analysis and processing tasks execute on streaming data in a dynamic, resource-constrained environment; and algorithms that are scalable and intelligent are needed to exploit the high spatial resolution and temporal fidelity of in situ data. However, limitations imposed by the coexistence of multiple in situ tasks, sequential data access, and the removal of human interaction from in situ workflows also complicate algorithm design. Portable algorithms that deliver peak performance are needed for both in situ and post hoc execution over multiple computational platforms in order to realize efficient utilization of each computational platform, maximize programmer productivity, and facilitate software maintenance.

New Research Directions In situ resource constraints motivate algorithms that use approximate and reduced representations, low-rank methods, functional approximations, and the combination of low- and high-fidelity surrogate models. Moreover, the accuracy and quality requirements and guarantees of approximate approaches need to be quantified and validated in the context of the workflow.

Research is required to modify existing post hoc algorithms and develop new in situ algorithms to satisfy the needs of modern use cases on emerging system architectures that can feature massive scale, many cores, deep memory hierarchies, or embedded lightweight edge devices. Examples of such algorithms include reduced representations and low-rank approximations Austin et al. (2016), statistical Hazarika et al. (2018); Thompson et al. (2011); Biswas et al. (2018); Dutta et al. (2017), topological Morozov and Weber (2013, 2014); Gyulassy et al. (2012, 2019); Landge et al. (2014), wavelets Li et al. (2017); Salloum et al. (2018), compression Di and Cappello (2016); Lindstrom (2014); Brislawn et al. (2012), and feature detection Guo et al. (2017) methods. Surrogate models and multifidelity models can be geometric Peterka et al. (2018); Nashed et al. (2019), statistical Lawrence et al. (2017); Lohrmann et al. (2017), or neural network He et al. (2019). Required are performance-portable algorithms that can be productively deployed by scientists for analyzing real-time, noisy, streaming data from physical experiments or sensors, in conjunction with traditional simulation models.

Algorithms for in situ analysis of multimodal and streaming data are needed. In addition to traditional simulations, data may originate from experiments, observations, multiphysics simulations, and/or ensemble workflows. Outputs will be increasingly high dimensional and multifidelity, will have uncertainties that must be accounted, and may be available only in a streaming fashion. Specific research directions include rewriting algorithms for streaming data (single-pass, multipass, and sliding window), dynamic scheduling of multiple data sources, and integrating heterogeneous data representations.

New classes of AI algorithms are needed to augment traditional visualization, topological, and statistical analyses. Examples include graph analytics (e.g., clustering, community detection) and semi- or unsupervised methods featuring reinforcement learning or transfer learning. Although these methods exist in other contexts, redesigning AI algorithms for scientific in situ analysis is crucial because existing algorithms that rely on massive amounts of training data and require long training times cannot be run, in their current state, in situ. Further research is needed to assess whether data analysis models base on AI can be trusted to automate ISDM decision-making.

Potential benefits In situ algorithms reduce time to solution compared with post hoc, mainly by avoiding I/O roundtrips to/from disk storage. In situ analysis also provides unprecedented access to every value computed by a simulation or generated by an experiment, provided that algorithms are scalable and intelligent and can utilize this capability to the fullest extent possible. Intermediate data products computed in memory can also elicit new types of analyses on those data and enhance the power of scientific analysis. The ability to assimilate multimodal and multiscale data in situ could provide a comprehensive view of scientific phenomena.

Controllable ISDM

Understand the design space of autonomous decision-making and control of in situ workflows.

Key questions What metrics best describe the ISDM design space? How can that space be defined, codified, and evaluated to support design decision-making and control? How can we dynamically adjust the organization, placement, and utilization of data to improve performance and satisfy user requirements?

Research opportunities Decisions concerning scheduling and placement of computations, choosing data structures and algorithms, planning for reliability, and satisfying user requirements create a complex design space for ISDM. Understanding that space is crucial to making intelligent design decisions, both by humans and autonomously, and the capability to optimize a constrained ISDM design space will enable predictable performance and scientific validity. Codifying the design space and developing metrics and benchmarks to evaluate it will also promote sharing of metrics and parameters across communities.

New Research Directions Research is needed to develop rigorous, explainable, reliable, and trustworthy decision-making in ISDM software. The complexity, nonlinearity, and dynamism of in situ workflows mandate augmenting or replacing human control with autonomous control. Autonomous control would enable real-time feedback and fine tuning of the ISDM components, potentially at a much higher frequency and accuracy than possible by a human.

Optimization, whether static or dynamic, of ISDM parameters requires codifying the metrics that best describe the design space. Researchers need to agree on a common taxonomy and language for describing ISDM. A set of community-wide benchmarks and test suites (i.e., miniapps for canonical workflow problems) is needed. Only then can the results of various optimization and control strategies be meaningfully compared.

Methods are needed for incorporating user intent (constraints or hints) in the ISDM design space. There are four aspects of incorporating constraints: specification, recording, execution, and provenance. Constraints may be specified by using service-level agreements (SLAs) or quality-of-service (QoS) contracts via the programming model. A record of the constraints needs to be stored in the system somewhere, for example, as part of the data model. The ISDM framework then needs to execute the constraints in its communication and execution models, and the extent to which the constraints could be honored needs to be recorded in the provenance of the execution and linked to the scientific results.

Both challenges and opportunities exist in understanding and incorporating the tradeoffs between quality of service and quality of results into the control of ISDM. Measuring or theoretically deriving the effect of algorithmic and operational parameters on performance (e.g., time to solution) and data quality (e.g., amount of error introduced) is a necessary step. Mapping data quality to scientific quality (e.g., confidence interval) is also required, but this is an open problem. Research is needed to understand this relationship.

Potential benefits Automating workflows can save human and machine resources because scientists can focus on domain-specific challenges, and computing platforms can be used more efficiently than with human control based on trial and error. Capturing the right information automatically about the workflow can lead to improved understanding of the workflow performance and of the science results. The development of performance models and other standardized metrics to evaluate the control operations will promote the sharing and reuse of information and the training of computer and domain scientists.

Composable ISDM

Develop interoperable ISDM components and capabilities for an agile and sustainable programming paradigm.

Key questions Can the composition of ISDM software components maximize programmer productivity and usability? What design decisions of ISDM software components promote their interoperability in order to ensure the long-term utility of ISDM software for the science community? How can we eliminate the burden on users wanting to transition current analysis and processing methods from post hoc to in situ?

Research opportunities There is an opportunity to design modular ISDM software using best engineering practices, combined with a clear focus on science mission needs. Realizing this opportunity requires long-lived sustainable ISDM frameworks that are adopted and used by the science community because they are able to compose in situ workflows from modular interoperable building blocks. The flexible composition of interoperable ISDM software components will enable developers and end-users to choose from an array of widely available tools, thereby increasing productivity, portability, and usability, resulting in agile and reusable software.

New Research Directions The first step is to develop community guidelines for broadly accepted specifications for ISDM software. Representative use cases for ISDM should be enumerated to drive software requirements and needs. The goal of such an activity is to define a minimal set of community guidelines for interfaces and data models to promote interoperability. Such guidelines must strike the right balance between commonality, generality, and extensibility.

Continued research is needed to design middleware to shield users from implementation differences and to provide platform portability. This middleware should be designed based on the lightweight independent functional decomposition of software pieces, at multiple granularities, from very coarse (entire applications) to very fine (single-purpose libraries). The decomposition should allow the ISDM community to exploit third-party software tools from other software ecosystems.

ISDM software must employ strict quality assurance and testing. Component-level testing, testing at scale, performance analysis and characterization, and multistage testing of entire workflows over a distributed area are all needed. Lacking are ISDM workflow test benchmark suites and sufficient performance models (analytical and/or empirical) to predict expected behavior. Also needed are deployment strategies for packaging and delivering software over combinations of operating system and run-time versions, compilers, and software dependencies.

Research should target working with science facilities (computational and experimental) and industry partners to develop, deploy, and support ISDM software over an extended lifetime of up to 20 years. Deeper engagement with science facilities is necessary in order to deploy software, define use-policies (such as on-demand job queues or network connections with remote facilities), and provide outreach and training for users. Long-term sustainability also requires industry partnerships to deploy ISDM in production software stacks, to provide software testing and maintenance, and to support new hardware over a longer period of time beyond the initial R&D activities.

Potential benefits Increasing the number and breadth of analysis tools to bring to bear on science problems would be a direct result of interoperable and composable ISDM software. Encouraging users to rely on reusable ISDM software frameworks developed and maintained for general use would provide measurable cost savings because individual application teams would not have to reinvent software infrastructure that is needed across the board. Such a model requires, however, that ISDM infrastructure be easy to use, robust, and sustainable. Given appropriate research in the design of interoperable software components coupled with support through computing facilities and industry partnerships, ISDM software would avail application scientists of all the advantages that the in situ computing model offers.

Transparent ISDM

Increase confidence in reproducible science, deliver repeatable performance, and discover new data features through the provenance of ISDM.

Key questions How can provenance and metadata support data interpretability, discovery, reuse, and reproducibility of results? How can these artifacts be captured automatically and analyzed in situ, at scale?

Research opportunities The ability to capture and query provenance and metadata in situ will support reproducibility and replicability, post hoc analysis, data discovery, and performance diagnostics. A recent report National Academies of Sciences, Engineering (2019) defines reproducibility as obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis. Replicability, a broader concept, implies obtaining consistent results across studies performed by different teams answering the same scientific question, each using its own data.

Provenance and metadata are needed for both reproducibility and replicability. In situ provenance is crucial to understanding scientific results, assessing correctness, and connecting underlying models and algorithms with workflow execution. The capability to capture and analyze provenance data within the time and space requirements of the domain science can both improve ISDM performance and ensure scientific validity. Also needed is understanding and quantifying the data uncertainties in the underlying computational models and algorithms, in particular understanding how uncertainties are compounded and propagated by multiple in situ tasks.

New Research Directions Scalable and portable provenance capture is needed for ISDM applications. Scalable compression of provenance data can augment judicious selection; hence, developing compression methods tailored for provenance is another potential research avenue.

New algorithms are needed to analyze provenance in situ and enable real-time control. In situ provenance processing is necessary in order to provide decision support for ISDM frameworks and the tasks running in them: for example, to tune the execution parameters or handle anomalies in real time.

Provenance will need to be integrated from multiple sources. Multimodal science use cases will generate multisource provenance, and these provenance data will need to be assimilated in a similar fashion as scientific data from multiple sources. The successful development and deployment of such tools will depend on the existence of standardized formats and libraries to capture provenance across numerous domain sciences.

New research is needed to understand what, if any, biases are introduced by in situ algorithms, both for processing scientific data and for provenance data. Traditional approaches to scientific validation and reproduction will have to be re-evaluated in light of the dynamic decisions made by in situ tasks that perform function approximation, analysis, or experiment design. Specialized provenance techniques that capture relevant information about the analysis method and that quantify uncertainty are needed.

Provenance systems should be co-designed with computational platforms and integrated with system-level provenance systems, as well as with standards and components developed in other communities. ISDM provenance systems will need to interact with these platform subsystems. Integrating application-level data with system data can potentially yield new functionality and utility.

Progress in ISDM provenance research would also benefit from recognizing work related to digital data preservation, data access, and provenance performed in other communities, such as libraries and digital archives.

Potential benefits Understanding the provenance of ISDM can lead to predictable and repeatable execution of workflows, ultimately reducing data size and shortening time to solution. Efficient capture and in situ analysis of provenance information would also increase confidence in scientific conclusions. Workflows that are validated through the provenance of data would be more trustworthy, reusable, and explainable than those whose data lineage is unknown.

Conclusions

Scientific computing will increasingly incorporate a number of different tasks that need to be managed along with computational models or experiments—ensemble analysis, data-driven science, artificial intelligence, machine learning, surrogate modeling, and graph analytics—all nontraditional applications unheard of in HPC a few years ago. Many of these tasks will need to execute concurrently, that is, in situ, with simulations and experiments sharing the same computing resources.

There are two primary, interdependent motivations for processing and managing data in situ. The first motivation is that the in situ methodology enables scientific discovery from a broad range of data sources—HPC simulations, experiments, scientific instruments, and sensor networks—over a wide scale of computing platforms including leadership-class HPC, clusters, clouds, workstations, and embedded devices at the edge. The second motivation is the need to decrease data volumes. ISDM can make critical contributions to managing large data volumes from computations and experiments, with the aim of minimizing data movement, saving storage space, and boosting resource efficiency—often while simultaneously increasing scientific precision.

Six PRDs highlight the components and capabilities needed for ISDM to be successful for a wide variety of applications: making ISDM capabilities more pervasive, controllable, composable, and transparent, with a focus on greater coordination with the software stack, and a diversity of fundamentally new data algorithms.

- **Pervasive ISDM:** Apply ISDM methodologies and in situ workflows on a variety of platforms and scales.
- **Co-designed ISDM:** Coordinate the development of ISDM with the underlying system software so that it is part of the software stack.
- **In Situ Algorithms:** Redesign data analysis algorithms for the in situ paradigm.
- **Controllable ISDM:** Understand the design space of autonomous decision-making and control of in situ workflows.
- **Composable ISDM:** Develop interoperable ISDM components and capabilities for an agile and sustainable programming paradigm.
- **Transparent ISDM:** Increase confidence in reproducible science, deliver repeatable performance, and discover new data features through the provenance of ISDM.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Argonne National Laboratory is managed by UChicago Argonne, LLC, for the U.S. Department of Energy Office of Science under contract DE-AC02-06CH11357. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This research also used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. Los Alamos National Laboratory is managed by Triad National Security, LLC for the U.S Dept. of Energy's NNSA under contract 89233218CNA000001. This manuscript has been authored in part by employees of Brookhaven Science Associates, LLC under Contract No. DESC0012704. Oak Ridge National Laboratory is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). This work was also supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Acknowledgements

The authors would like to thank all workshop participants for their contributions: Jim Ahrens, Ann Almgren, Katie Antypas, Edmun Begoli, Amber Boehnlein, Ron Brightwell, Jacqueline Chen, Hank Childs, Warren Davis, Marc Day, Jai Dayal, Ewa Deelman, Lei Ding, Nicola Ferrier, Fernanda Foertter, Berk Geveci, Katrin Heitmann, Bruce Hendrickson, Jay Hnilo, Adolphy Hoisie, Dan Jacobson, Shantenu Jha, Ming Jiang, Kirk Jordan, Scott Klasky, Kerstin Kleese van Dam, Eric Lancon, Earl Lawrence, Steve Legensky, Jay Lofstead, Andrew Lumsdaine, Kwan-Liu Ma, Pat McCormick, Ken Moreland, Dmitriy Morozov, H. Sarp Oral, Manish Parashar, Valerio Pascucci, Amedeo Perazzo, Beth Plale, Thomas Proffen, Mike Ringenburg, Silvio Rizzi, Rob Ross, Tim Scheibe, David Schissel, Malachi Schram, Katie Schuman, Nicholas Schwarz, Han-Wei Shen, Eric Stephan, Victoria Stodden, Michela Taufer, Justin Wozniak, John Wu, and Hongfeng Yu.

References

Agarwal S, Quach TT, Parekh O, Hsia AH, DeBenedictis EP, James CD, Marinella MJ and Aimone JB (2016) Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and Its Application to Sparse Coding. *Frontiers in Neuroscience* 9: 484. DOI:10.3389/fnins.2015.00484. URL https://www.frontiersin.org/article/10.3389/fnins.2015.00484.

Agrawal K, Fahey MR, McLay R and James D (2014) User Environment Tracking and Problem Detection with XALT. In: *Proceedings of the First International Workshop on HPC User Support Tools*. IEEE Press, pp. 32–40. DOI:10.1109/HUST.2014.6.

Allan D, Caswell T, Campbell S and Rakitin M (2019) Bluesky's Ahead: A Multi-Facility Collaboration for an a la Carte Software Project for Data Acquisition and Management. *Synchrotron Radiation News* 32: 19–22. DOI:10.1080/08940886.2019.1608121.

Ames S, Gokhale M and Maltzahn C (2013) QMDS: A File System Metadata Management Service Supporting a Graph Data Model-Based Query Language. *International Journal of Parallel, Emergent and Distributed Systems* 28(2): 159–183.

Antcheva I, Ballintijn M, Bellenot B, Biskup M, Brun R, Buncic N, Canal P, Casadei D, Couet O, Fine V, Franco L, Ganis G, Gheata A, Maline DG, Goto M, Iwaszkiewicz J, Kreshuk A, Segura DM, Maunder R, Moneta L, Naumann A, Offermann E, Onuchin V, Panacek S, Rademakers F, Russo P and Tadel M (2009) ROOT - a C++ Framework for Petabyte Data Storage, Statistical Analysis and Visualization. *Computer Physics Communications* 180(12): 2499–2512. DOI:https://doi.org/10.1016/j.cpc.2009.08.005. URL http://www.sciencedirect.com/science/article/pii/S0010465509002550. 40 YEARS OF CPC: A celebratory issue focused on quality software for high performance, grid and novel computing architectures.

Argonne Leadership Computing Facility (2019) Singularity. https://www.alcf.anl.gov/user-guides/singularity.

Austin W, Ballard G and Kolda TG (2016) Parallel Tensor Compression for Large-Scale Scientific Data. In: *Proceedings of 2016 IEEE international parallel and distributed processing symposium (IPDPS)*. IEEE, pp. 912–922.

Ayachit U (2015) *The ParaView Guide: A Parallel Visualization Application*. USA: Kitware, Inc. ISBN 1930934300, 9781930934306.

Ayachit U, Bauer AC, Geveci B, O'Leary P, Moreland K, Fabian N and Mauldin J (2015) ParaView Catalyst: Enabling In Situ Data Analysis and Visualization. In: Weber GH (ed.) *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, ISAV 2015, Austin, TX, USA, November 15-20, 2015*. ACM. ISBN 978-1-4503-4003-8, pp. 25–29.

Ayachit U, Whitlock B, Wolf M, Loring B, Geveci B, Lonie D and Bethel EW (2016) The SENSEI Generic In Situ Interface. In: *Proceedings of In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV 2016)*. Salt Lake City, UT, USA. DOI:http://doi.org/10.1109/ISAV.2016.13. LBNL-1007263.

Babuji Y, Woodard A, Li Z, Katz DS, Clifford B, Kumar R, Lacinski L, Chard R, Wozniak J, Foster I, Wilde M and Chard K (2019) Parsl: Pervasive Parallel Programming in Python. In: *28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*. DOI:10.1145/3307681.3325400. URL https://doi.org/10.1145/3307681.3325400.

Balay S, Brown J, Knepley M, McInnes LC and Smith B (2016) Providing Mixed-Language and Legacy Support within a Library: Experiences of Developing PETSc. Chapter accepted for publication in the upcoming book on *Software Engineering for Science*, editors J. Carver, H. C. Hong, and G. Thiruvathukal.

Bartlett RA, Demeshko I, Gamblin T, Hammond G, Heroux MA, Johnson J, Klinvex A, Li X, McInnes LC, Moulton JD, Osei-Kuffuor D, Sarich J, Smith B, Willenbring JM and Yang UM (2017) xSDK Foundations: Toward an Extreme-Scale Scientific Software Development Kit. *CoRR* abs/1702.08425. URL http://arxiv.org/abs/1702.08425.

Bauer AC, Abbasi H, Ahrens J, Childs H, Geveci B, Klasky S, Moreland K, O'Leary P, Vishwanath V, Whitlock B et al. (2016) In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms. In: *Computer Graphics Forum*, volume 35. Wiley Online Library, pp. 577–597.

Benedict MN, Mundy MB, Henry CS, Chia N and Price ND (2014) Likelihood-based Gene Annotations for Gap Filling and Quality Assessment in Genome-scale Metabolic Models. *PLoS Comput Biol* 10(10): e1003882.

Bennett JC, Bhagatwala A, Chen JH, Pinar A, Salloum M and Seshadhri C (2016) Trigger Detection for Adaptive Scientific Workflows Using Percentile Sampling. *SIAM J. Scientific Computing* 38(5).

Bethel EW, Jacobsen J and Holland P (1994) Site Remediation in a Virtual Environment. In: Moorhead RJ, Silver DE and Uselton SP (eds.) *Visual Data Exploration and Analysis, Proceedings of SPIE 2178*. San Jose CA, USA, pp. 78–87. LBNL-34865.

Bicer T, Gürsoy D, De Andrade V, Kettimuthu R, Scullin W, De Carlo F and Foster IT (2017) Trace: A High-Throughput Tomographic Reconstruction Engine for Large-Scale Datasets. *Advanced structural and chemical imaging* 3(1): 6.

Biswas A, Dutta S, Pulido J and Ahrens J (2018) In Situ Data-Driven Adaptive Sampling for Large-Scale Simulation Data Summarization. In: *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV '18. Dallas, Texas: ACM. ISBN 978-1-4503-6579-6, pp. 13–18. DOI:10.1145/3281464.3281467. URL http://doi.acm.org/10.1145/3281464.3281467https://datascience.dsscale.org/wp-content/uploads/sites/3/2019/01/LA-UR-18-28035.pdf.

Biswas A, Dutta S, Shen HW and Woodring J (2013) An Information-Aware Framework for Exploring Multivariate Data Sets. *IEEE Transactions on Visualization and Computer Graphics* 19(12): 2683–2692.

Brightwell R, Oldfield R, Maccabe AB and Bernholdt DE (2013) Hobbes: Composition and Virtualization as the Foundations of an Extreme-Scale OS/R. In: *Proceedings of the 3rd International Workshop on Runtime and Operating Systems for Supercomputers*, ROSS '13. New York, NY, USA: ACM. ISBN 978-1-4503-2146-4, pp. 2:1–2:8. DOI:10.1145/2491661.2481427. URL http://doi.acm.org/10.1145/2491661.2481427.

Brislawn CM, Woodring J, Mniszewski S, DeMarle D and Ahrens J (2012) Subband Coding for Large-Scale Scientific Simulation Data Using JPEG 2000. In: *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*. IEEE, pp. 201–204. URL https://datascience.dsscale.org/wp-content/uploads/sites/3/2016/06/SubbandCodingForLarge-ScaleScientificSimulationData pdf. LA-UR-12-1352.

Caino-Lores S, Carretero J, Nicolae B, Yildiz O and Peterka T (2018) Spark-DIY: A Framework for Interoperable Spark Operations with High Performance Block-Based Data Models. In: *Proceedings of IEEE BDCAT*. IEEE.

Carns P, Jenkins J, Cranor CD, Atchley S, Seo S, Snyder S and Ross RB (2016) Enabling NVM for Data-Intensive Scientific Services. In: *4th Workshop on Interactions of NVM/Flash with Operating Systems and Workloads INFLOW 16)*.

Center for Open Science (2011) Open Science Framework. https://osf.io.

Childs H, Brugger E, Whitlock B, Meredith J, Ahern S, Pugmire D, Biagas K, Miller M, Weber GH, Krishnan

H, Fogal T, Sanderson A, Garth C, Bethel EW, Camp D, Rübel O, Durant M, Favre J and Navratil P (2012) VisIt: An End-User Tool for Visualizing and Analyzing Very Large Data. In: Bethel EW, Childs H and Hansen C (eds.) *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, Chapman & Hall, CRC Computational Science. Boca Raton, FL, USA: CRC Press/Francis–Taylor Group. ISBN 9781439875728, pp. 357–372. URL http://www.crcpress.com/product/isbn/9781439875728. LBNL-6320E.

Chirigati F, Shasha D and Freire J (2013) ReproZip: Using Provenance to Support Computational Reproducibility. In: *Presented as part of the 5th USENIX Workshop on the Theory and Practice of Provenance*.

Choi JY, Wu K, Wu JC, Sim A, Liu QG, Wolf M, Chang C and Klasky S (2013) ICEE: Wide-Area In Transit Data Processing Framework for Near Real-Time Scientific Applications. In: *Proceedings of Big Data Analytics: Challenges and Opportunities (BDAC-13).*

Dai D, Ross RB, Carns P, Kimpe D and Chen Y (2014) Using Property Graphs for Rich Metadata Management in HPC Systems. In: *2014 9th Parallel Data Storage Workshop*. IEEE, pp. 7–12.

Damiani D, Dubrovin M, Gaponenko I, Kroeger W, Lane TJ, Mitra A, O'Grady CP, Salnikov A, Sanchez-Gonzalez A, Schneider D and Yoon CH (2016) Linac Coherent Light Source Data Analysis Using *psana*. *Journal of Applied Crystallography* 49(2): 672–679. DOI:10.1107/S1600576716004349. URL https://doi.org/10.1107/S1600576716004349.

Dart E, Rotman L, Tierney B, Hester M and Zurawski J (2014) The Science DMZ: A Network Design Pattern for Data-Intensive Science. *Scientific Programming* 22(2): 173–185.

Daurer BJ, Krishnan H, Perciano T, Maia FRNC, Shapiro DA, Sethian JA and Marchesini S (2017) Nanosurveyor: A Framework for Real-Time Data Processing. *Advanced Structural and Chemical Imaging* 3(1): 7. DOI:10.1186/s40679-017-0039-0. URL https://doi.org/10.1186/s40679-017-0039-0.

Davis E, Zender CS, Arctur DK, O'Brien K, Jelenak A, Santek D, Dixon MJ, Whiteaker TL and Yang K (2017) NetCDF-CF: Supporting Earth System Science with Data Access, Analysis, and Visualization. *AGU Fall Meeting Abstracts* .

Deelman E, Gannon D, Shields M and Taylor I (2009) Workflows and e-Science: An Overview of Workflow System Features and Capabilities. *Future Generation Computer Systems* 25(5): 528–540.

Deelman E, Peterka T, Altintas I, Carothers C, van Dam KK, Moreland K, Parashar M, Ramakrishnan L, Taufer M and Vetter J (2016) The Future of Scientific Workflows Report of the DOE NGNS/CS Scientific Workflows Workshop. Technical report. http://science.energy.gov/~/media/ascr/pdf/programdocuments/docs/workflows_final_report.pdf.

Deelman E, Peterka T, Altintas I, Carothers C, van Dam KK, Moreland K, Parashar M, Ramakrishnan L, Taufer M and Vetter J (2017) The Future of Scientific Workflows. *International Journal of High Performance Computing Applications* 32(1): 159–175.

Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ, Mayani R, Chen W, Ferreira da Silva R, Livny M and Wenger K (2015) Pegasus: A Workflow Management System for Science Automation. *Future Generation Computer Systems* 46: 17–35. DOI:10.1016/j.future.2014.10.008. URL http://pegasus.isi.edu/publications/2014/2014-fgcs-deelman.pdf.

Di S and Cappello F (2016) Fast Error-Bounded Lossy HPC Data Compression with SZ. In: *Parallel and Distributed Processing Symposium, 2016 IEEE International*. IEEE, pp. 730–739.

Dong B, Byna S, Wu K, , Johansen H, Johnson JN and Keen N (2016) Data Elevator: Low-Contention Data Movement in Hierarchical Storage System. In: *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. pp. 152–161. DOI:10.1109/HiPC.2016.026.

Dorier M, Antoniu G, Cappello F, Snir M, Sisneros R, Yildiz O, Ibrahim S, Peterka T and Orf L (2016) Damaris: Addressing Performance Variability in Data Management for Post-Petascale Simulations. *ACM Trans. Parallel Comput.* 3(3): 15:1–15:43.

Dorier M, Carns P, Harms K, Latham R, Ross R, Snyder S, Wozniak J, Gutiérrez S, Robey B, Settlemyer B et al. (2018) Methodology for the Rapid Development of Scalable HPC Data Services. In: *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*. IEEE, pp. 76–87.

Dorier M, Dreher M, Peterka T and Ross R (2017) CoSS: Proposing a Contract-Based Storage System for HPC. In: *Proceedings of the 2nd Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems*. ACM, pp. 13–18.

Dreher M, Perarnau S, Peterka T, Iskra K and Beckman P (2017) In Situ Workflows at Exascale: System Software to the Rescue. In: *Proceedings of ISAV SC17 Workshop*.

Dreher M and Peterka T (2016) Bredala: Semantic Data Redistribution for In Situ Applications. In: *Proceedings of IEEE Cluster 2016*. IEEE.

Dreher M and Peterka T (2017) Decaf: Decoupled Dataflows for In Situ High-Performance Workflows. Technical report, ANL/MCS-TM-371, Argonne National Lab., Argonne, IL, USA.

Dutta S, Shen HW and Chen JP (2018) In Situ Prediction Driven Feature Analysis in Jet Engine Simulations. In: *PacificVis*. IEEE Computer Society. ISBN 978-1-5386-1424-2, pp. 66–75. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8365785.

Dutta S, Woodring J, Shen HW, Chen JP and Ahrens J (2017) Homogeneity Guided Probabilistic Data Summaries for Analysis and Visualization of Large-Scale Data Sets. In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. pp. 111–120. DOI:10.1109/PACIFICVIS.2017.8031585. URL https://datascience.dsscale.org/wp-content/uploads/sites/3/2018/08/la-ur_18-27370.pdf. LA-UR-18-27370.

Fahey M, Jones N, Hadri B and Hitchcock B (2010) The Automatic Library Tracking Database. *Proceedings of the Cray User Group* .

Folk M, Cheng A and Yates K (1999) HDF5: A File Format and I/O Library for High Performance Computing Applications. In: *Proceedings of Supercomputing*, volume 99. pp. 5–33.

Friedman-Hill E, Hoffman E, Gibson M and Clay R (2015) Incorporating Workflow for V&V/UQ in the Sandia Analysis

Workbench. Technical Report SAND2015-0425C, Sandia National Laboratories.

Gamblin T, LeGendre MP, Collette MR, Lee GL, Moody A, de Supinski BR and Futral WS (2015) The Spack Package Manager: Bringing order to HPC Software Chaos. In: *Supercomputing 2015 (SC'15)*. Austin, Texas. URL http://tgamblin.github.io/pubs/spack-sc15.pdf. LLNL-CONF-669890.

Gao T, Guo Y, Zhang B, Cicotti P, Lu Y, Balaji P and Taufer M (2017) Mimir: Memory-Efficient and Scalable MapReduce for Large Supercomputing Systems. In: *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, pp. 1098–1108.

Gittens A, Rothauge K, Wang S, Mahoney MW, Kottalam J, Gerhardt L, Ringenburg M and Maschhoff K (2018) Alchemist: An Apache Spark - MPI Interface. *Concurrency and Computation: Practice and Experience* .

Graves R, Jordan TH, Callaghan S, Deelman E, Field E, Juve G, Kesselman C, Maechling P, Mehta G, Milner K et al. (2011) CyberShake: A Physics-based Seismic Hazard Model for Southern California. *Pure and Applied Geophysics* 168(3-4): 367–381.

Guo H, Peterka T and Glatz A (2017) In Situ Magnetic Flux Vortex Visualization in Time-Dependent Ginzburg-Landau Superconductor Simulations. In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, pp. 71–80.

Gyulassy A, Bremer PT and Pascucci V (2019) Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy. *IEEE Trans. Vis. Comput. Graph* 25(1): 1183–1192. URL http://doi.ieeecomputersociety.org/10.1109/TVCG.2018.2864848.

Gyulassy A, Pascucci V, Peterka T and Ross RB (2012) The Parallel Computation of Morse-Smale Complexes. In: *Proc. IEEE International Parallel and Distributed Processing Symposium (26th IPDPS'12)*. Shanghai, China: IEEE Computer Society, pp. 484–495.

Hazarika S, Biswas A and Shen H (2018) Uncertainty Visualization Using Copula-Based Analysis in Mixed Distribution Models. *IEEE Transactions on Visualization and Computer Graphics* 24(1): 934–943.

He W, Wang J, Guo H, Wang KC, Shen HW, Raj M, Nashed YS and Peterka T (2019) InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations. *IEEE Transactions on Visualization and Computer Graphics* 26(1): 23–33.

Henseler D, Landsteiner B, Petesch D, Wright C and Wright NJ (2016) In: *Cray User Group (CUG)*.

Hildreth MD, Boehnlein A, Cranmer K, Dallmeier S, Gardner R, Hacker T, Heinrich L, Jimenez I, Kane M, Katz DS, Malik T, Maltzahn C, Neubauer M, Neubert S, Pivarski J, Sexton E, Shiers J, Simko T, Smith S, South D, Verbytskyi A, Watts G and Wozniak J (2018) HEP Software Foundation Community White Paper Working Group – Data and Software Preservation to Enable Reuse. *arXiv e-prints* : arXiv:1810.01191.

Hjelm NT and Wright C (2017) Libhio: Optimizing IO on Cray XC Systems with DataWarp. In: *Cray User Group*.

Interlandi M, Shah K, Tetali SD, Gulzar MA, Yoo S, Kim M, Millstein T and Condie T (2015) Titian: Data Provenance Support in Spark. *Proceedings of the VLDB Endowment* 9(3): 216–227.

Jain A, Ong SP, Chen W, Medasani B, Qu X, Kocher M, Brafman M, Petretto G, Rignanese GM, Hautier G, Gunter D and Persson KA (2015) FireWorks: A Dynamic Workflow System Designed for High-Throughput Applications. *Concurrency and Computation: Practice and Experience* 27(17): 5037–5059. DOI:10.1002/cpe.3505. URL http://dx.doi.org/10.1002/cpe.3505. CPE-14-0307.R2.

James CD, Aimone JB, Miner NE, Vineyard CM, Rothganger FH, Carlson KD, Mulder SA, Draelos TJ, Faust A, Marinella MJ, Naegle JH and Plimpton SJ (2017) A Historical Survey of Algorithms and Hardware Architectures for Neural-Inspired and Neuromorphic Computing Applications. *Biologically Inspired Cognitive Architectures* 19: 49–64. DOI:https://doi.org/10.1016/j.bica.2016.11.002. URL http://www.sciencedirect.com/science/article/pii/S2212683X16300561.

Jenkins J, Shipman G, Mohd-Yusof J, Barros K, Carns P and Ross R (2017) A Case Study in Computational Caching Microservices for HPC. In: *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, pp. 1309–1316.

Joubert W, Weighill DA, Kainer D, Climer S, Justice A, Fagnan K and Jacobson D (2018) Attacking the Opioid Epidemic: Determining the Epistatic and Pleiotropic Genetic Architectures for Chronic Pain and Opioid Addiction. In: *Proceedings of SC18*. IEEE / ACM, pp. 57:1–57:14. URL http://dl.acm.org/citation.cfm?id=3291656.

Khan F, Hammonds J, Narayanan S et al. (2013) Effective End-to-End Management of Data Acquisition and Analysis for X-ray Photon Correlation Spectroscopy. In: *Presented at the ICALEPCS. Available at: http://accelconf. web. cern. ch/AccelConf/ICALEPCS2013*.

Kincade K (2015) Shifter. https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/2015/shifter-makes-container-based-hpc-a-breeze/.

Knpfer A, Rssel C, an Mey D, Biersdorff S, Diethelm K, Eschweiler D, Geimer M, Gerndt M, Lorenz D and Malony A (2012) Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir. In: *Tools for High Performance Computing 2011*. Springer, pp. 79–91. DOI:10.1007/978-3-642-31476-6_7.

Koerner LJ, Caswell TA, Allan DB and Campbell SI (2019) A Python Instrument Control and Data Acquisition Suite for Reproducible Research. *IEEE Transactions on Instrumentation and Measurement* PP: 1–1. DOI:10.1109/TIM.2019.2914711.

Kurth T, Treichler S, Romero J, Mudigonda M, Luehr N, Phillips EH, Mahesh A, Matheson M, Deslippe J, Fatica M, Prabhat and Houston M (2018) Exascale Deep Learning for Climate Analytics. In: *Proceedings of SC18*. IEEE / ACM, pp. 51:1–51:12. URL http://dl.acm.org/citation.cfm?id=3291656.

Kutare M, Eisenhauer G, Wang C, Schwan K, Talwar V and Wolf M (2010) Monalytics: Online Monitoring and Analytics for Managing Large Scale Data Centers. In: *Proceedings of the 7th international conference on Autonomic computing*. pp. 141–150.

Lammel S, Zahn F and Frning H (2016) SONAR: Automated Communication Characterization for HPC Applications. In:

Taufer M, Mohr B and Kunkel JM (eds.) *High Performance Computing*, Lecture Notes in Computer Science. Springer International Publishing. ISBN 978-3-319-46079-6, pp. 98–114.

Landge AG, Pascucci V, Gyulassy A, Bennett J, Kolla H, Chen J and Bremer PT (2014) In-Situ Feature Extraction of Large Scale Combustion Simulations Using Segmented Merge Trees. In: Damkroger T and Dongarra JJ (eds.) *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2014, New Orleans, LA, USA, November 16-21, 2014*. IEEE Computer Society. ISBN 978-1-4799-5500-8, pp. 1020–1031. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7012142; http://dl.acm.org/citation.cfm?id=2683593.

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: ACM. ISBN 978-1-4503-5139-3, pp. 42–46. DOI:10.1145/3144769.3144778. URL http://doi.acm.org/10.1145/3144769.3144778.

Larsen M, Woods A, Marsaglia N, Biswas A, Dutta S, Harrison C and Childs H (2018) A Flexible System for in Situ Triggers. In: Wolf M, Moreland K and Bethel EW (eds.) *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, ISAV@SC 2018, Dallas, TX, USA, November 11, 2018*. ACM. ISBN 978-1-4503-6579-6, pp. 1–6.

Lawrence E, Heitmann K, Kwan J, Upadhye A, Bingham D, Habib S, Higdon D, Pope A, Finkel H and Frontiere N (2017) The Mira-Titan universe, II: Matter Power Spectrum Emulation. *The Astrophysical Journal* 847(1): 50.

Li S, Sane S, Orf L, Mininni P, Clyne J and Childs H (2017) Spatiotemporal Wavelet Compression for Visualization of Scientific Simulation Data. In: *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. pp. 216–227.

Lindstrom P (2014) Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics* 20(12): 2674–2683.

Liu Q, Logan J, Tian Y, Abbasi H, Podhorszki N, Choi JY, Klasky S, Tchoua R, Lofstead J, Oldfield R, Parashar M, Samatova N, Schwan K, Shoshani A, Wolf M, Wu K and Yu W (2014) Hello ADIOS: The Challenges and Lessons of Developing Leadership Class I/O Frameworks. *Concurrency and Computation: Practice and Experience* 26(7): 1453–1473. DOI:10.1002/cpe.3125. URL http://doi.wiley.com/10.1002/cpe.3125.

Lockwood GK, Wright NJ, Snyder S, Carns P, Brown G and Harms K (2018) TOKIO on ClusterStor: Connecting Standard Tools to Enable Holistic I/O Performance Analysis .

Lofstead JF, Klasky S, Schwan K, Podhorszki N and Jin C (2008) Flexible IO and Integration for Scientific Codes Through the Adaptable IO System ADIOS). In: *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '08. New York, NY, USA: ACM. ISBN 978-1-60558-156-9, pp. 15–24.

Lohrmann E, Luki Z, Morozov D and Mller J (2017) Programmable In Situ System for Iterative Workflows. *Proceedings of Workshop on Job Scheduling Strategies for Parallel Processing* : 122–131.

Magini N, Di Girolamo A, Guan W, Wenaus T, Tsulaia V, Lassnig M, Walker R, Maeno T and Nilsson P (2018) Towards an Event Streaming Service for ATLAS Sata Processing. Technical Report ATL-SOFT-PROC-2018-050, CERN, Geneva. URL https://cds.cern.ch/record/2649518.

Malitsky N, Chaudhary A, Jourdain S, Cowan M, OLeary P, Hanwell M and Van Dam KK (2017) Building Near-Real-Time Processing Pipelines with the Spark-MPI Platform. In: *2017 New York Scientific Data Summit (NYSDS)*. IEEE, pp. 1–8.

McInnes L (2019) IDEAS. https://ideas-productivity.org/ideas-classic.

Mommessin C, Dreher M, Raffin B and Peterka T (2017) Automatic Data Filtering for In Situ Workflows. In: *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, pp. 370–378.

Morozov D and Weber G (2013) Distributed Merge Trees. In: *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '13. New York, NY, USA: ACM. ISBN 978-1-4503-1922-5, pp. 93–102.

Morozov D and Weber GH (2014) Distributed Contour Trees. In: *Topological Methods in Data Analysis and Visualization III*. Springer International Publishing, pp. 89–102.

Nashed Y, Peterka T, Mahadevan V and Grindeanu I (2019) Rational Approximation of Scientific Data. In: *Proceedings of 2019 International Conference on Computational Science*.

National Academies of Sciences, Engineering (2019) *Reproducibility and Replicability in Science*. ISBN 978-0-309-48613-2. DOI:10.17226/25303.

National Energy Research Scientific Computing Center (2019) Science Gateways. https://docs.nersc.gov/services/science-gateways.

NSLS-II Data Acquisition and Management Group (2019) NSLS-II Bluesky Documentation. https://nsls-ii.github.io/architecture-overview.html.

Oak Ridge Leadership Computing Facility (2019) Oak Ridge Leadership Computing Facility. https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit.

OMG Object Management Group (2000) Notification Service Specification 1.0. ftp://www.omg.org/pub/doc/formal/00-06-20.pdf.

Pandolfi RJ, Allan DB, Arenholz E, Barroso-Luque L, Campbell SI, Caswell TA, Blair A, De Carlo F, Fackler S, Fournier AP, Freychet G, Fukuto M, Gürsoy D, Jiang Z, Krishnan H, Kumar D, Kline RJ, Li R, Liman C, Marchesini S, Mehta A, N'Diaye AT, Parkinson DY, Parks H, Pellouchoud LA, Perciano T, Ren F, Sahoo S, Strzalka J, Sunday D, Tassone CJ, Ushizima D, Venkatakrishnan S, Yager KG, Zwart P, Sethian JA and Hexemer A (2018) *Xi-cam: A Versatile Interface for Data Visualization and Analysis*. *Journal of Synchrotron Radiation* 25(4): 1261–1270. DOI:10.1107/S1600577518005787. URL https://doi.org/10.1107/S1600577518005787.

Parker SG and Johnson CR (1995) SCIRun: A Scientific Programming Environment for Computational Steering. In: *Proceedings of Supercomputing'95*. San Diego, CA: ACM/IEEE.

Pasquier T, Han X, Moyer T, Bates A, Hermant O, Eyers D, Bacon J and Seltzer M (2018) Runtime Analysis of Whole-System Provenance. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, pp. 1601–1616.

Perarnau S, Gupta R and Beckman P (2013) Argo: An Exascale Operating System and Runtime. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC15*.

Peterka T, Bard D, Bennett J, Bethel EW, Oldfield R, Pouchard L, Sweeney C and Wolf M (2019) ASCR Workshop on In Situ Data Management: Enabling Scientific Discovery from Diverse Data Sources. Technical report. DOI:10.2172/1493245.

Peterka T, Nashed Y, Grindeanu I, Mahadevan V, Yeh R and Trixoche X (2018) Foundations of Multivariate Functional Approximation for Scientific Data. In: *Proceedings of 2018 IEEE Symposium on Large Data Analysis and Visualization*.

Pouchard L, Huck K, Matyasfalvi G, Tao D, Tang L, Van Dam H and Yoo S (2018) Prescriptive Provenance for Streaming Analysis of Workflows at Scale. In: *2018 New York Scientific Data Summit (NYSDS)*. IEEE. DOI:10.1109/NYSDS.2018.8538951.

Pouchard LC, Poole S, Lothian J and Groer C (2009) ORNL TM 145: Open Standards for Sensor Information Processing. Technical report, Oak Ridge National Laboratory.

Salloum M, Bennett JC, Pinar A, Bhagatwala A and Chen JH (2015) Enabling Adaptive Scientific Workflows via Trigger Detection. *ISAV2015 Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization* abs/1508.04731. URL http://arxiv.org/abs/1508.04731.

Salloum M, Fabian N, Hensinger DM, Lee J, Allendorf EM, Bhagatwala A, Blaylock ML, Chen JH, Templeton JA and Tezaur I (2018) Optimal Compressed Sensing and Reconstruction of Unstructured Mesh Datasets. *Data Science and Engineering* 3(1): 1–23.

Sanderson A, Humphrey A, Schmidt J and Sisneros R (2018) Coupling the Uintah Framework and the VisIt Toolkit for Parallel In Situ Data Analysis and Visualization and Computational Steering. In: *Proceedings of International Conference on High Performance Computing*. Springer, pp. 201–214.

Schuman CD, Potok TE, Patton RM, Birdwell JD, Dean ME, Rose GS and Plank JS (2017) A Survey of Neuromorphic Computing and Neural Networks in Hardware. *CoRR* abs/1705.06963. URL http://arxiv.org/abs/1705.06963.

Severa W, Parekh O, Carlson KD, James CD and Aimone JB (2016) Spiking Network Algorithms for Scientific Computing. In: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. pp. 1–8. DOI:10.1109/ICRC.2016.7738681.

Shende SS and Malony AD (2006) The TAU Parallel Performance System. *The International Journal of High Performance Computing Applications* 20(2): 287–311. DOI:10.1177/1094342006064482.

Shipman G, Campbell S, Dillow D, Doucet M, Kohl J, Granroth G, Miller R, Stansberry D, Proffen T and Taylor R (2014) Accelerating Data Acquisition, Reduction, and Analysis at the Spallation Neutron Source. In: *Proceedings of 2014 IEEE 10th International Conference on eScience, eScience 2014*, volume 1. pp. 223–230. DOI:10.1109/eScience.2014.31.

Singh A, Stephan E, Elsethagen T, MacDuff M, Raju B, Schram M, van Dam KK, Kerbyson DJ and Altintas I (2016) Leveraging Large Sensor Streams for Robust Cloud Control. In: *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 2115–2120. DOI:10.1109/BigData.2016.7840839.

Snyder S, Carns P, Harms K, Ross R, Lockwood GK and Wright NJ (2016) Modular HPC I/O Characterization with Darshan. In: *2016 5th Workshop on Extreme-Scale Programming Tools (ESPT)*. IEEE, pp. 9–17.

Srinivasan R (1995) XDR: External Data Representation Standard. Technical report. DOI:10.17487/RFC1832. URL https://rfc-editor.org/rfc/rfc1832.txt.

Supercomputing Conference (2018) SC18 Home Page. https://sc18.supercomputing.org.

Tallent N, Mellor-Crummey J, Adhianto L, Fagan M and Krentel M (2008) HPCToolkit: Performance Tools for Scientific Computing. In: *Journal of Physics: Conference Series*, volume 125. IOP Publishing, p. 012088. DOI:10.1088/1742-6596/125/1/012088.

Thompson DC, Levine JA, Bennett J, Bremer PT, Gyulassy A, Pascucci V and Pébay PP (2011) Analysis of Large-Scale Scalar Data Using Hixels. In: Rogers DH and Silva CT (eds.) *LDAV*. IEEE Computer Society. ISBN 978-1-4673-0156-5, pp. 23–30. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6083212.

Wang C and Shen HW (2011) Information Theory in Scientific Visualization. *Entropy* 13(1): 254–273.

Wang T, Oral S, Wang Y, Settlemyer B, Atchley S and Yu W (2014) BurstMem: A High-Performance Burst Buffer System for Scientific Applications. In: *2014 IEEE International Conference on Big Data (Big Data)*. pp. 71–79. DOI:10.1109/BigData.2014.7004215.

Wang Y, Agrawal G, Bicer T and Jiang W (2015) Smart: A MapReduce-like Framework for In-Situ Scientific Analytics. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, p. 51.

Weber GH, Dillard SE, Carr HA, Pascucci V and Hamann B (2007) Topology-Controlled Volume Rendering. *IEEE Trans. Vis. Comput. Graph* 13(2): 330–341. URL http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.47.

Widener P, Eisenhauer G, Schwan K and Bustamante FE (2002) Open Metadata Formats: Efficient XML-Based Communication for High Performance Computing. *Cluster Computing* 5(3): 315–324.

Wood C, Sane S, Ellsworth D, Gimenez A, Huck K, Gamblin T and Malony A (2016) A Scalable Observation System for Introspection and In Situ Analytics. In: *2016 5th Workshop on Extreme-Scale Programming Tools (ESPT)*. IEEE, pp. 42–49.

Woodring J, Ahrens J, Figg J, Wendelberger J, Habib S and Heitmann K (2011) In-Situ Sampling of a Large-Scale Particle Simulation for Interactive Visualization and Analysis. In: *Computer Graphics Forum*, volume 30. Wiley Online Library, pp. 1151–1160. URL https://datascience.dsscale.org/wp-content/uploads/sites/3/2016/06/In-situSamplingOfALarge-ScaleParticleSimulationForI pdf. LA-UR-11-02106.

World Wide Web Consortium Working Group (2013) PROV Overview: An Overview of the PROV Family of Documents. https://www.w3.org/TR/prov-overview.

Wozniak JM, Jain R, Balaprakash P, Ozik J, Collier NT, Bauer J, Xia F, Brettin TS, Stevens R, Mohd-Yusof J, Garcia-Cardona C, Essen BV and Baughman M (2018) CANDLE/Supervisor: A Workflow Framework for Machine Learning Applied to Cancer Research. *BMC Bioinformatics* 19-S(18): 59–69.

xSDK Project (2019) xSDK. https://xsdk.info.

Yildiz O, Ejarque J, Chan H, Sankaranarayanan S, Badia R and Peterka T (2019) Heterogeneous Hierarchical Workflow Composition. *IEEE Computing in Science and Engineering Special Issue on Scientific Workflows* .

Yu J and Buyya R (2005) A Taxonomy of Workflow Management Systems for Grid Computing. *Journal of Grid Computing* 3(3-4): 171–200.

Zhang X, Abbasi H, Huck K and Malony AD (2016) Wowmon: A Machine Learning-Based Profiler for Self-Adaptive Instrumentation of Scientific Workflows. *Procedia Computer Science* 80: 1507–1518.

Author Biographies

Tom Peterka is a computer scientist at Argonne National Laboratory, scientist at the University of Chicago Consortium for Advanced Science and Engineering (CASE), adjunct assistant professor at the University of Illinois at Chicago, and fellow of the Northwestern Argonne Institute for Science and Engineering (NAISE). His research interests are in large-scale parallel in situ analysis of scientific data. Recipient of the 2017 DOE early career award and four best paper awards, Peterka has published in ACM SIGGRAPH, IEEE VR, IEEE TVCG, and ACM/IEEE SC, among other top conferences and journals. Peterka received his Ph.D. in computer science from the University of Illinois at Chicago in 2007, and he currently works actively in several DOE- and NSF-funded projects.

Deborah Bard is a computing systems engineer at the National Energy Research Scientific Computing center at Lawrence Berkeley National Lab, where she leads the Data Science Engagement Group. Her career spans research in particle physics, cosmology and computing, including publications in PRL, PRD, MNRAS, AAS, Astronomy and Computing and ACM/IEEE SC. She obtained her Ph.D. at Edinburgh University, and worked at Imperial College London and SLAC National Accelerator Laboratory before joining NERSC, where her group supports supercomputing for experimental science. Her work currently focuses on data-intensive computing, including machine learning at scale.

Janine C. Bennett is the Manager for the Extreme-Scale Data Science and Analytics department at Sandia National Laboratories. Janine joined Sandia in 2009, with a research focus on developing topological and statistical data analysis techniques for science and engineering applications. Over the years, her work has evolved to include a broader systems-level emphasis, encompassing in situ analysis on extreme-scale HPC platforms as well as next generation programming models and runtime system research. She has proposed and led projects funded by DOE NNSA ASC, DOE SC ASCR, and Sandia LDRD programs and regularly serves on multiple international program and steering committees. Prior to joining Sandia, Janine was a Lawrence Scholar in the Center for Applied Scientific Computing at LLNL. Janine holds a B.S. in Computer Science and Engineering, and an M.S. and Ph.D. in Computer Science from the University of California, Davis.

E. Wes Bethel is an innovator, a visionary, an entrepreneur, a mentor, a data and computer scientist, a technologist, and a leader. His technical background in computer and data science spans several fields, which lie at the nexus of computing and data: high performance computing, scientific/information visualization, graphics, scientific data/image analysis, and machine learning. Over the course of his 30+ years in the field, he has been instrumental in advancing the field of high performance scientific visualization and analytics through the R&D of new methods and software tools, and applying them with success to many fields of science. He is passionate about enabling scientific knowledge discovery through the convergence of data and computing, a pursuit that requires ongoing evolution and adaptation to a changing technological and scientific landscape. He is an ACM Distinguished Scientist, a Senior Fellow at the Berkeley Institute for Data Science (UC Berkeley), an Adjunct Associate Professor of Computer Science at Old Dominion University, and is a member of ACM (SIGGRAPH, SIGHPC) and IEEE. He earned his Ph.D. in Computer Science from the University of California, Davis.

Ron A. Oldfield manages the Scalable Analysis and Visualization department in the Center for Computing Research at Sandia National Laboratories in Albuquerque, NM. He received a B.Sc. in Computer Science from the University of New Mexico in 1993, and a Ph.D. in Computer Science from Dartmouth College in 2003. In his 20-year career at SNL, Ron has been an active contributor and principal investigator in areas such as large-scale seismic imaging, HPC system software, scalable I/O (file systems and libraries), resilience, system simulation, performance modeling, scalable analytics, and visualization. Since 2007, he has had programmatic roles as a project manager for Sandia projects in I/O, Data Management, Scalable Analysis and Visualization, and Machine Learning for the NNSA Advanced Scientific Computing (ASC) program and the Exascale Computing Project. He has more than 40 peer-reviewed publications and is a member of IEEE and ACM.

Line Pouchard is a Computational Scientist who joined the Center for Data-Driven Discovery in the Computational Science Initiative at Brookhaven National Laboratory as Senior Researcher in January 2017. Prior positions include Assistant Professor at Purdue University, Staff Scientist at Oak Ridge National Laboratory, Research Associate Professor and Advisory Board, School of Information Sciences, University of Tennessee, Knoxville, co-founder and lead of the DataONE Integration and Semantics Working Group. She leads multi-disciplinary efforts improving scientific data discovery, data management and curation in scientific domains such as Earth and Environmental Sciences, Ecology, Materials Science, Extreme Scale Computing, Structural Engineering, and National Security. Her research focuses on provenance, Semantic Web technologies, and text mining for big data. She has a Ph.D. from the Graduate Center of the City University of New York, and an M.S. from the University of Tennessee, Knoxville.

Christine Sweeney is a computer scientist at Los Alamos National Laboratory in the Applied Computer Science group. Her research interests are in in situ data analysis and workflows for experimental science and also in core computer science areas such as parallel programming models, compilers and domain-specific languages. She currently works actively on Exascale Computing Project ExaFEL and ExaLearn projects, LANL LDRD projects and IARPA projects, all involving workflows for light source experimental science data analytics. She also organizes workshops in support of data analytics for light source science at light source user meetings as well as for LANL.

Matthew Wolf is a Senior Research Scientist at Oak Ridge National Laboratory in the Scientific Data Group. His background in Mathematics and Computational Physics as well as High Performance Computing shapes much of his research interest in computer operating systems, middleware, and CS education. His research targets high performance, scalable applications, particularly focused on I/O and adaptive event middlewares. Specific research topics include adaptive I/O interfaces, metadata-rich data services, creation of dynamic, semantic indexes for scientific data, and handling and fusion of heterogeneous data types.