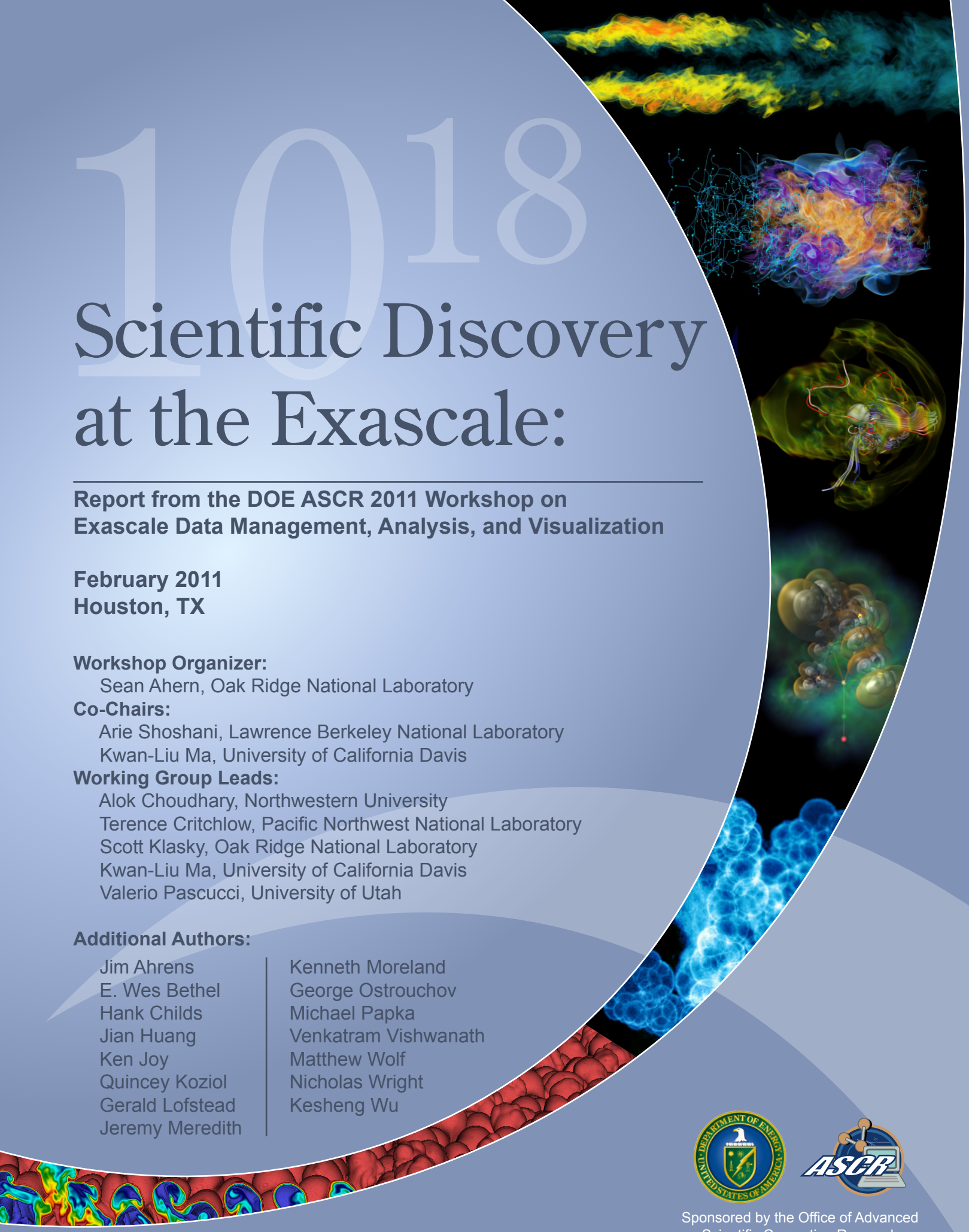

2018 Scientific Discovery at the Exascale:

Report from the DOE ASCR 2011 Workshop on Exascale Data Management, Analysis, and Visualization

February 2011
Houston, TX

Workshop Organizer:
Sean Ahern, Oak Ridge National Laboratory

Co-Chairs:
Arie Shoshani, Lawrence Berkeley National Laboratory
Kwan-Liu Ma, University of California Davis

Working Group Leads:
Alok Choudhary, Northwestern University
Terence Critchlow, Pacific Northwest National Laboratory
Scott Klasky, Oak Ridge National Laboratory
Kwan-Liu Ma, University of California Davis
Valerio Pascucci, University of Utah

Additional Authors:

Jim Ahrens
E. Wes Bethel
Hank Childs
Jian Huang
Ken Joy
Quincey Koziol
Gerald Lofstead
Jeremy Meredith
Kenneth Moreland
George Ostrouchov
Michael Papka
Venkatram Vishwanath
Matthew Wolf
Nicholas Wright
Kesheng Wu

Sponsored by the Office of Advanced Scientific Computing Research

Acknowledgements

DOE ASCR Workshop on Exascale Data Management, Analysis, and Visualization was held in Houston, TX, on 22-23 February 2011.

Support for the workshop web site was provided by Brian Gajus of ORNL. Kathy Jones and Deborah Counce from ORNL helped edit the report. Nathan Galli from the SCI Institute at the University of Utah designed the report cover. Sherry Hempfling and Angela Beach from ORNL helped with the workshop organization and provided support throughout the workshop.

Within DOE ASCR, Lucy Nowell provided valuable advice on the program and on participants.

The workshop organizers would like to thank the workshop attendees for their time and participation. We are indebted to the breakout group leaders: Alok Choudhary, Terence Critchlow, Scott Klasky, Kwan-Liu Ma, and Valerio Pascucci. We extend a special thanks to the attendees who served as additional authors of this report:

Jim Ahrens, Los Alamos National Laboratory
E. Wes Bethel, Lawrence Berkeley National Laboratory
Hank Childs, Lawrence Berkeley National Laboratory
Jian Huang, University of Tennessee
Ken Joy, University of California Davis
Quincey Koziol, The HDF Group
Gerald Lofstead, Sandia National Laboratory
Jeremy Meredith, Oak Ridge National Laboratory
Kenneth Moreland, Sandia National Laboratory
George Ostrouchov, Oak Ridge National Laboratory
Michael Papka, Argonne National Laboratory
Venkatram Vishwanath, Argonne National Laboratory
Matthew Wolf, Georgia Tech
Nicholas Wright, Lawrence Berkeley National Laboratory
Kesheng Wu, Lawrence Berkeley National Laboratory

Sean Ahern
8 July 2011

Contents

1 Executive Summary

In February 2011, the Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) convened a workshop to explore the problem of scientific understanding of data from High Performance Computation (HPC) at the exascale. The goal of this workshop report is to identify the research and production directions that the Data Management, Analysis, and Visualization (DMAV) community must take to enable scientific discovery for HPC as it approaches the exascale (1 quintillion floating point calculations per second = 1 exaflop). Projections from the international TOP500 list [20] place that date around 2018–2019.

Extracting scientific insight from large HPC facilities is of crucial importance for the nation. The scientific simulations that run on the supercomputers are only half of the "science"; scientific advances are made only once the data produced by the simulations is processed into an output that is understandable by an application scientist. As mathematician Richard Hamming famously said, "The purpose of computing is insight, not numbers." [29] It is precisely the DMAV community that provides the algorithms, research, and tools to enable that critical insight.

The hardware and software changes that will occur as HPC enters the exascale era will be dramatic and disruptive. Not only are scientific simulations forecasted to grow by many orders of magnitude, but also current methods by which HPC systems are programmed and data are extracted are not expected to survive into the exascale. Changing the fundamental methods by which scientific understanding is obtained from HPC simulations is a daunting task. Specifically, dramatic changes to on-node concurrency, access to memory hierarchies, accelerator and GPGPU processing, and input/output (I/O) subsystems will all necessitate reformulating existing DMAV algorithms and workflows. Additionally, reductions in inter-node and off-machine communication bandwidth will require rethinking how best to provide scalable algorithms for scientific understanding.

Principal Finding: The disruptive changes imposed by a progressive movement toward the exascale in HPC threaten to derail the scientific discovery process. Today's successes in extracting knowledge from large HPC simulation output are not generally applicable to the exascale era, and simply scaling existing techniques to higher concurrency is insufficient to meet the challenge.

Recommendation: Focused and concerted efforts toward co-designing processes for exascale scientific understanding must be adopted by DOE ASCR. These efforts must include research into effective in situ processing frameworks, new I/O middleware systems, fine-grained visualization and analysis algorithms to exploit future architectures, and co-scheduling analysis and simulation on HPC platforms. Such efforts must be pursued in direct collaboration with the application domain scientists targeting exascale architectures. To ensure effective delivery to scientists, DMAV researchers require access to "testbed" systems so that they can prototype and pilot effective solutions.

Our full set of findings and recommendations may be found in Section 8 "Conclusion: Findings and Recommendations" on page 31.

2 Prior Work

Work toward extreme-scale data understanding has occurred for many years. The research roadmap outlined in this workshop report builds upon decades of prior work done by the DMAV community and others. The two primary programmatic efforts within DOE that have been fertile ground for advances in scientific understanding have been the National Nuclear Security Administration's (NNSA) Advanced Simulation and Computing (ASC) program and the two phases of the Scientific Discovery through Advanced Computing (SciDAC) program of the Office of Science. These two programs have advanced parallel data analysis, visualization, and scientific data management into the petascale era though their direct support of NNSA and ASCR mission objectives. For a detailed look at how these two programs have dramatically affected scientific discovery, see Appendix A on page 36.

3 Workshop and Report Overview

The two-day exascale workshop was broken into two major activities, information gathering and collaborative exchange, each on a separate day. Information gathering consisted of presentations by experts in the field. For a full list of workshop participants, please see Appendix B on page 38.

First, Andy White of Los Alamos National Laboratory and Stephen Poole of Oak Ridge National Laboratory (ORNL) gave presentations on expected **exascale hardware and system architectures**.

Then a select group of application scientists presented their computational domains with an eye toward the scientific understanding challenges they expect at the exascale. Gary Strand, National Center for Atmospheric Research, summarized the needs of the computational **climate** community. Bronson Messer of ORNL described the needs of physicists simulating astrophysics, particularly **core-collapse supernovae**. Jackie Chen of Sandia National Laboratories discussed the needs of the computational **combustion** community and presented successful in situ frameworks at the petascale. C-S Chang, New York University, outlined the data understanding needs of the computational **fusion** community.

The participants rounded out the day with presentations from members of the DMAV community, who discussed research directions with potential for meeting the needs of the application scientists as exascale computing approaches. Hank Childs of Lawrence Berkeley National Laboratory (LBNL) presented techniques for scalable **visualization**. Scott Klasky of ORNL presented his work on scalable **I/O infrastructures** and techniques for automating **scientific workflow** processes. John Wu, LBNL, summarized the latest techniques for **data indexing** and I/O acceleration. Nagiza Samatova of North Carolina State University presented techniques for enhancing scientific datasets through the general framework of **scalable analytics**.

The workshop continued on the second day with two sessions of breakout groups to promote collaborative exchange. The first session featured three groups:

- **Concurrent Processing & In Situ** led by Kwan-Liu Ma of the University of California–Davis
- **I/O and Storage** led by Scott Klasky of ORNL
- **Data Postprocessing** led by Alok Choudhary of Northwestern University

This was followed by a second session with two breakout groups:

- **Visualization and Analysis** led by Valerio Pascucci of the University of Utah
- **Scientific Data Management** led by Terence Critchlow of Pacific Northwest National Laboratory

Each breakout group presented its findings and recommendations to the plenary session in the afternoon as the workshop closed out.

In this workshop report, we attempt to follow the same format as the workshop itself. We first present the architectural changes expected as we progress to the exascale (Section 4 below). We present a summary of direct application needs (Section 5 on page 4). We then outline our recommended research roadmap (Section 6 on page 15), roughly broken into the categories of the five breakout groups of the workshop. We round out the report with identified areas for co-design and collaboration (Section 7 on page 30) and conclude with our findings and recommendations (Section 8 on page 31).

4 Future Architectures – Overview and Implications

The most significant changes at the exascale come from architectural changes in the underlying hardware platforms. In the 1980s and 1990s, researchers could count on a Moore's Law doubling of scalar floating performance every 18 months; and in the 2000s, the HPC community saw sizable gains in scaling distributed memory execution as the nodes maintained a fairly standard balance between memory, I/O, and CPU performance. The exascale, however, will see an age of significant imbalances between system components. These imbalances, detailed below, will necessitate a sea change in how computational scientists exploit HPC hardware.

4.1 Exascale System Architecture

Potential exascale system architecture parameters are shown in Table 1. The table provides projected numbers for the design of two "swim lanes" of hardware design representing radically different design choices. Also shown are the equivalent metrics for a machine today and the difference from today's machines.

Table 1: Expected Exascale Architecture Parameters and Comparison with Current Hardware (from "The Scientific Grand Challenges Workshop: Architectures and Technology for Extreme Scale Computing" [55]).

		"2018"		
System Parameter	2011	Swim Lane 1	Swim Lane 2	Factor Change
System Peak	2 Pf/s	1 Ef/s		500
Power	6 MW	≤20 MW		3
System Memory	0.3 PB	32–64 PB		100–200
Total Concurrency	225K	1B×10	1B×100	40,000–400,000
Node Performance	125 GF	1 TF	10 TF	8–80
Node Concurrency	12	1,000	10,000	83–830
Network BW	1.5 GB/s	100 GB/s	1000 GB/s	66–660
System Size (nodes)	18700	1,000,000	100,000	50–500
I/O Capacity	15 PB	300–1000 PB		20–67
I/O BW	0.2 TB/s	20–60 TB/s		10–30

From examining these differences it is clear that an exascale-era machine will not simply be a petascale machine scaled in every dimension by a factor of 1,000. The principal reason for this is the need to control the power usage of such a machine.

The implications for users of such systems are numerous:

- **Total concurrency in the applications must rise by a factor of about 40,000–400,000, but available memory will rise only by a factor of about 100.** From a scientist's perspective, the ratio of memory to compute capability is critical in determining the size of the problem that can be solved. The processor dictates how much computing can be done; the memory dictates the size of the problem that can be handled. The disparity of growth between computing and storage means that memory will become a much more dominant factor in the size of problem that can be solved, so applications cannot just scale to the speed of the machine. In other words, the current weak-scaling approaches will not work. Scientists and computer scientists will have to rethink how they are going to use the systems; the factor of >100 loss in memory per compute thread means that there will be a need to completely redesign current application codes, and the supporting visualization and analytics frameworks, to enable them to exploit parallelism as much as possible. It is also important to note that most of this parallelism will be on-node.

- **For both power and performance reasons, locality of data and computation will be much more important at the exascale.** On an exascale-class architecture, the most expensive operation, from both a power and performance perspective, will be moving data. The further the data is moved, the more expensive the process will be. Therefore, approaches that maximize locality as much as possible and pay close attention to their data movement are likely to be the most successful. As well as locality between nodes (horizontal locality), it will also be essential to pay attention to on-node locality (vertical locality), as the memory hierarchy is likely to get deeper. This also implies that synchronization will be very expensive, and the work required to manage synchronization will be high. Thus successful approaches will also minimize the amount of synchronization required.

- **The I/O storage subsystem of an exascale machine will be, relatively speaking, much smaller and slower compared with both the peak flops and the memory capacity.** From both an energy usage and a cost perspective, it seems likely that much less aggregate disk-based I/O capacity and bandwidth will be available on an exascale-class machine. Thus, both the storage of

simulation results and checkpointing for resiliency are likely to require new approaches. In fact, some part of the analysis of simulation results is likely to be performed in situ in order to minimize the amount of data written to permanent storage.

4.2 Potentially Disruptive New Technology — NVRAM

At the same time as these trends are occurring, new nonvolatile memory technologies are emerging that could somewhat mitigate the issues. Probably the most well known of these is NAND flash, because of its ubiquity in consumer devices such as phones and cameras. Today its usage is just beginning to be explored in the realm of HPC. Compared with a regular spinning disk, flash memory has a large latency advantage for both read and write operations. Therefore, for example, it has very attractive performance characteristics for small I/O operations [40]. Thus augmenting a node of a HPC machine with a NVRAM-based (nonvolatile random-access memory) device should make it possible to both improve I/O bandwidth for checkpointing and provide a potentially attractive technology for use as a swap device to extend memory capacity, potentially allowing us to partially mitigate both of the trends identified above. There are also other varieties of nonvolatile memory technologies beginning to emerge, such as phase change memory (PCM) and magnetic RAM (MRAM). These technologies all have different characteristics in terms of access times for read and write operations, as well as reliability, durability, cost, and so on. Users of exascale machines will have to exploit these new resources to maximize their performance as well as to improve their resiliency by using the NVRAM device as a storage resource for checkpointing.

5 User Needs and Use Cases

To appropriately guide the research roadmap for exascale computing, we need to ground ourselves in the specific and direct needs of the relevant DOE application communities, culling from their collective knowledge of their computational domains. We have identified commonalities among and across these groups, distilling prevalent use cases that can be considered cross-disciplinary.

5.1 Science Application Drivers

Representative exascale user requirements are drawn from the eight reports from the "Scientific Grand Challenges Workshop Series." Separate workshops were held for eight scientific domains, each attracting approximately 50 technical leaders in the field of extreme-scale computing. The resulting reports focus on the grand challenges of a specific scientific domain, the role for scientific computing in addressing those challenges, and actionable recommendations for overcoming the most formidable technical issues. Each report finds that significant increases in computing power are expected to lead to scientific breakthroughs in their field. In the summaries that follow, we focus primarily on the visualization, analysis, and data management requirements embedded in these reports. We distill some common themes across these reports in Section 5.2.

5.1.1 High-Energy Physics

The High-Energy Physics Workshop was held December 9–11, 2008, in Menlo Park, California. Co-chairs of the workshop were Roger Blandford of the SLAC National Accelerator Laboratory, Norman Christ of Columbia University, and Young-Kee Kim of Fermi National Accelerator Laboratory. The recommendations of that workshop are detailed in the report *Scientific Grand Challenges: Challenges for the Understanding the Quantum Universe and the Role of Computing at the Extreme Scale* [7].

The Grand Challenges report details the role of massive computation related to the needs of the high-energy physics community. The community sees large-scale computation as providing the framework within which research into the mysteries of the quantum universe can be undertaken. The requirements of this community are indeed extreme, enabling researchers to obtain increased accuracy in simulations, allowing the analysis and visualization of 100 petabyte data sets, and creating methods by which simulations can be optimized. To do these things, new software paradigms, numerical algorithms, and programming tools need to be developed to take advantage of the extreme-scale architectures that will be designed in the exascale. In

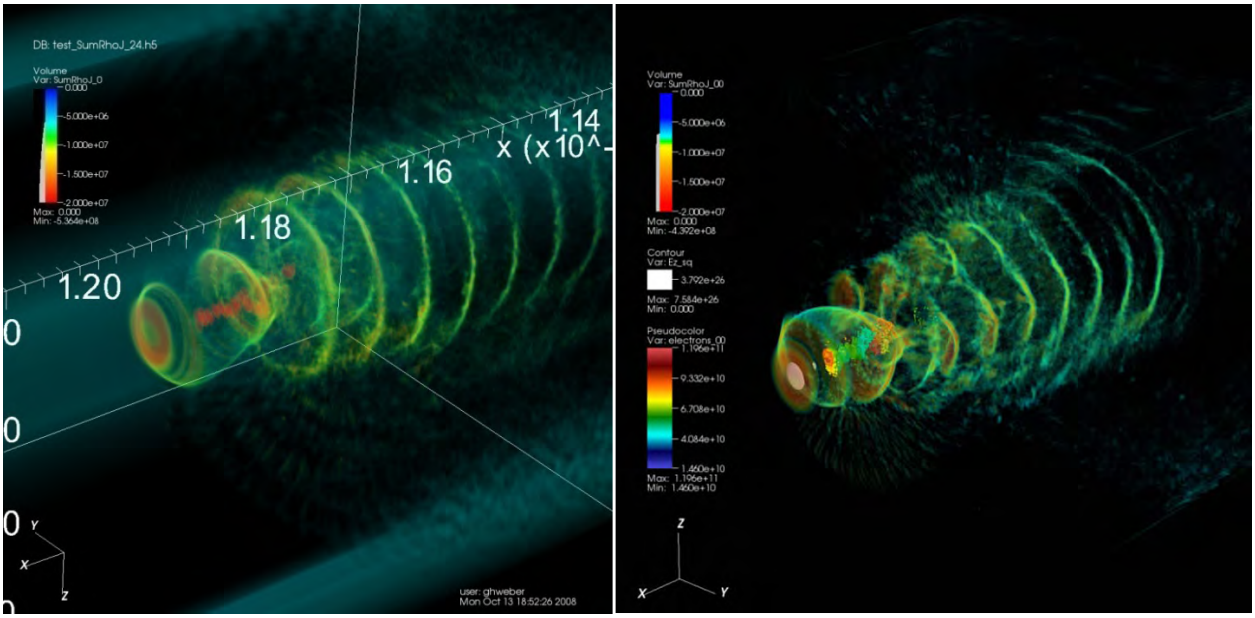

Figure 1: Two visualizations of an intense laser pulse (traveling from right to left) that is producing a density wake in a hydrogen plasma, shown by colored density isosurfaces in VisIt parallel visualizations of VORPAL simulations modeling LOASIS (LBNL) laser wakefield accelerator (LWFA) experiments. On the right, high-energy particles are overlaid with the wake, colored by their momentum, facilitating understanding of how these experiments produced narrow energy spread bunches for the first time in an LWFA (red). The present simulations run on 3,500 processors for 36 hours; and visualization of the 50 GB/time snapshot datasets runs on 32 processors, taking tens of minutes/snapshot. Future experiments will increase these demands by orders of magnitude. Data indexing has been shown to decrease time to discovery for these types of data sets by up to three orders of magnitude [73]. VORPAL is developed by the Tech-X Corporation, partially supported through the SciDAC accelerator modeling program (ComPASS). Image courtesy of David Bruhwiler (Tech-X Corporation). Image from the High-Energy Physics workshop report [7].

addition, the massive amounts of data developed and processed by these new systems will require end-to-end solutions for data management, analysis, and visualization, and the management and automation of the workflows associated with the simulations.

Cross-cutting issues are highly important to the high-energy physics community. These scientists recognize that their use of future exascale systems is predicated on the development of software tools that take advantage of architectural advances and the development of data exploration, analysis, and management tools that enable the discovery of new information in their data. Because current petascale platforms are architecturally ill suited to the task of massive data analysis, the community suggests that a data-intensive engine be developed (something with the total memory of an exascale computer, but fewer processor nodes and higher I/O bandwidth) for analysis of simulation results, observational data mining, and interactive visualization and data analysis.

The opportunities offered by exascale computing will enable the optimal design of large-scale instrumentation and simulations that go beyond the capabilities of today's systems. Cross-cutting issues are paramount in developing the capability to design these experiments and analyze the resulting data. Some specific findings of the workshop follow.

- The success of these simulation activities on new-generation extreme-scale computers requires advances in meshing, sparse-matrix algorithms, load balancing, higher-order embedded boundaries, optimization, data analysis, visualization, and fault tolerance.
- The understanding of highly nonlinear collective beam effects requires advances in particle-in-cell (PIC) technologies, pipelining algorithms, multi-language software infrastructure, data analysis, visualization, fault tolerance, and optimization. See Figure 1 for an example of beam simulation and visualization.

- Simulations in this field require the development of multiscale, multiphysics accelerator structure frameworks, including integrated electromagnetic, thermal, and mechanical analysis. To develop such a framework, advances are necessary in meshing, load balancing, solver, coupling technology (e.g., mesh to mesh), optimization, data analysis, and visualization (in situ).
- Success in many efforts requires advances in PIC methods, pipelining algorithms for quasi-static PIC models, multi-language software infrastructure, performance, data analysis, visualization, fault tolerance, dynamic load balancing, and mesh refinement. Improving the fidelity and scaling of reduced models will also require new algorithm development.
- The high-energy physics community recognizes that analysis and visualization of the resulting data will require major changes in the current file I/O paradigm. There is a need to develop the software infrastructure for physics analysis and visualization on the fly, in addition to enhancing the performance of the post-processing and postmortem analysis tools. It will be imperative to identify the problems and define the strategies to overcome I/O bottlenecks in applications running on many thousands to many hundreds of thousands of processors. In addition, a common data representation and well-defined interfaces are required to enable analysis and visualization.

The rewards of developing exascale systems for the high-energy physics community will be great, but the required efforts are also great. The simulation and analysis systems developed in the past will not transfer directly to the new generation of systems, and much work must be done to develop new methods that use the architectural advantages of these systems. Cross-cutting disciplines are extremely important here, as the development of new programming paradigms, numerical tools, analysis tools, visualization tools, and data management tools will dramatically impact the success of their research programs.

5.1.2 Climate

Computational climate science aims to develop physically and chemically accurate numerical models and their corresponding implementation in software. DOE's ASCR program develops and deploys computational and networking tools that enable scientists to model, simulate, analyze, and predict phenomena important to DOE. This community issued a report in 2008 [69], later summarized in 2010 [34], that sketches out how exascale computing would benefit climate science and makes the point that realizing those benefits requires significant advances in areas such as algorithm development for future architectures and DMAV.

Exascale computational capacity offers the potential for significant advances in several different aspects of climate science research. First, existing models presently run at ∼100 km grid resolution; yet accurate physical modeling of critical climate systems, such as clouds and their impact on radiation transfer within the atmosphere, require significantly higher spatial resolution. Some weather features, like hurricanes and cyclones, become "visible" in simulation output only as spatial resolution increases (Figure 2). Therefore, climate scientists expect evolution toward 1 km grid spacing, which by itself represents an increase of at least four orders of magnitude in the size of a single dataset.

Second, temporal resolution in the present typically consists of yearly summaries over a 1000 year period. It is likely to grow by one to two orders of magnitude as climate scientists pursue both decadal predictive capabilities, which require very high temporal resolution, and paleoclimate studies, which attempt to model climate changes that occur abruptly over the course of hundreds of centuries yet are missed in coarse temporal sampling (Figure 3). Related, accurate climate modeling requires graceful accommodation of multiscale phenomena: some processes—like the birth of cloud drops, ice crystals, and aerosols—occur over time scales of a few minutes but interact with larger-scale and longer-time circulation systems.

A third area for growth involves models that couple different components of climate, such as atmosphere, ocean, geochemistry, biochemistry, and aerosols. Such models represent a major potential advancement, since a climate model's predictive capability requires that it accurately model physics and chemistry within each of these regimes and accurately capture fluxes among them.

Fourth, these tools and their resulting data products will increasingly be used for prediction by a diverse set of stakeholders ranging from climate science researchers to policy makers.

Fifth, a collection of simulation runs can capture the variability that occurs across a range of different input conditions better than individual simulations can. Such ensemble collections play an important role in aiding understanding of the uncertainty, variability, and probability of potential climate changes.

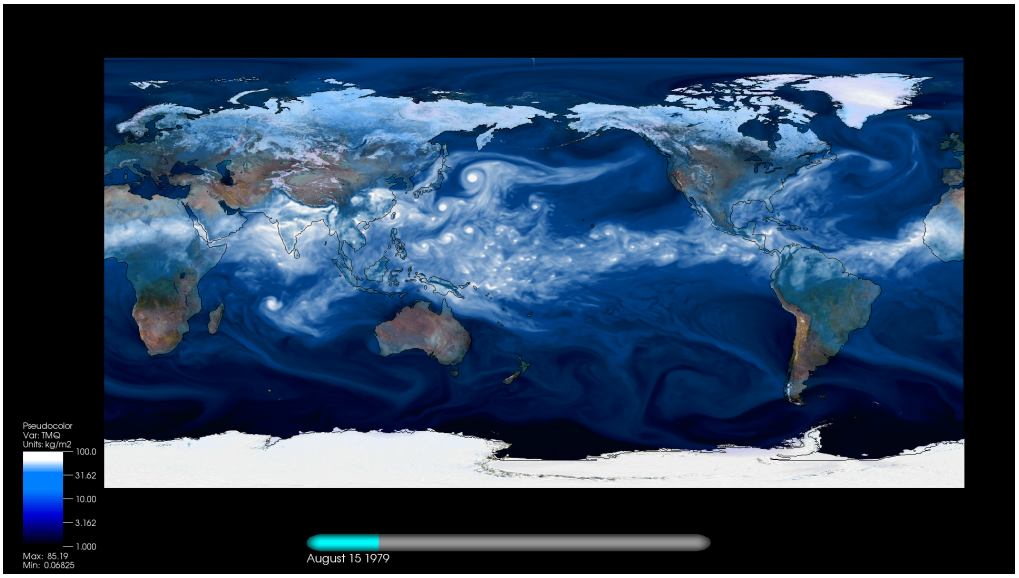

Figure 2: Global warming will probably change the statistics of tropical cyclones and hurricanes. In this high-resolution simulation, using the finite volume version of NCAR's Community Atmosphere Model (CAM-5), we study how well the model can reproduce observed tropical cyclone statistics. The simulated storms seen in this animation are generated spontaneously from the model's simulated weather conditions long after the initial conditions have been forgotten. The structure of these simulated tropical cyclones is surprisingly realistic, with the strongest storms rating as Category 4 on the Sapphir-Simpson scale. The image is a visualization of the total vertical column integrated water vapor. Simulation data by M. Wehner (LBNL), visualization by Prabhat (LBNL).

Finally, these software tools and their data products will ultimately be used by a large international community. The climate community expects that datasets, collectively, will range into the 100s of exabytes by 2020 [34, 69]. Because the simulation results are a product for subsequent analysis and model validation by a large international community, the in situ processing model, in which visualization/analysis is performed while simulation data is still resident in memory, is not a good match.

Distributed data. An important trend noted by the climate science reports is that an international community of climate science researchers are consumers of what will be hundreds of exabytes of data products. These products would be distributed over a wide geographic area in federated databases. Ideally, researchers would have a single, unified methodology for accessing such distributed data. The process of distributing the data will impose a substantial load on future networks, requiring advances not only in throughput/bandwidth but also in related technologies like monitoring and scheduling, movement/transmission, and space reservation. The explosive growth in data diversity will in turn require significant advances in metadata management to enable searches, and related subsetting capabilities so researchers can extract the portions of datasets of interest to them. It is likely that more visualization and analysis processing will happen "close to" the data, rather than by subsequent analysis of data downloaded to a researcher's remote machine.

Data size and complexity. Given the projected explosive growth in the size and complexity of datasets—four to six orders of magnitude for a single dataset—visualization and analysis technologies must evolve to accommodate larger and more complex data as input. Therefore, they must be able to take advantage of future machine architectures so they can leverage platforms with large amounts of memory and processing capacity. Also, seamless visualization or analysis of data from coupled models is relatively new; the existing software infrastructure is simply not designed to handle this complex new type of data.

Deployment of software and data. Given that future consumers of climate science data products will include a large community of climate scientists as well as non-experts like policy makers, a significant challenge is how to best approach dissemination of, and access to, data and software tools for analysis. An increasing amount of data analysis and visualization will probably, by necessity, be conducted "close to" the

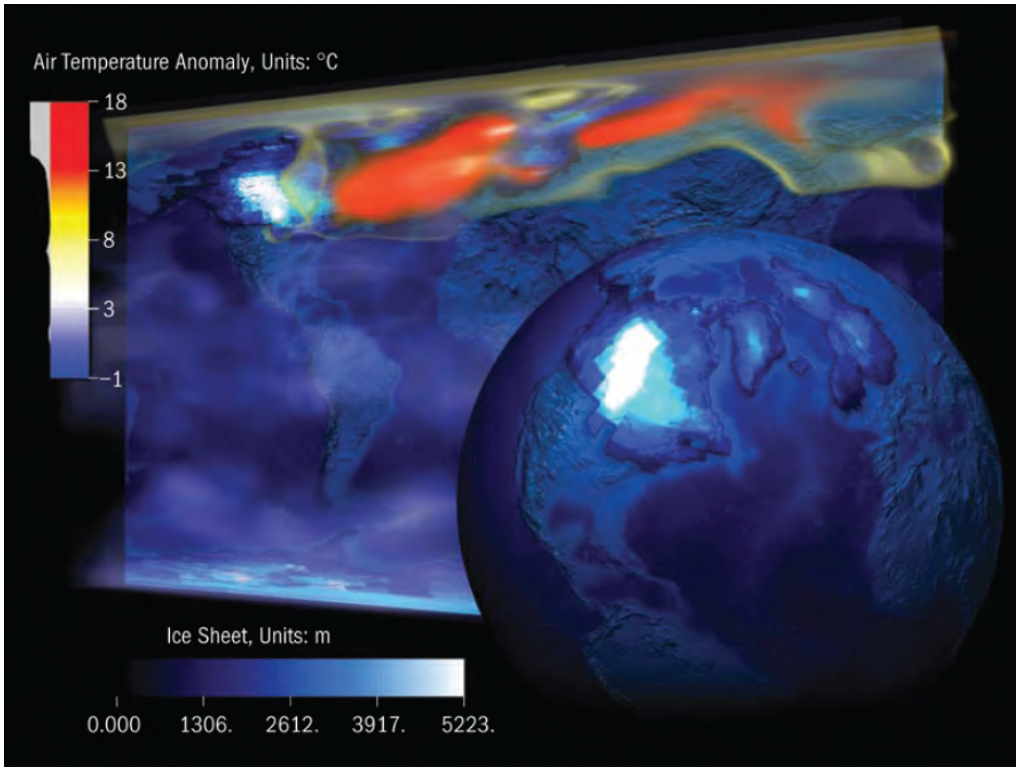

Figure 3: Abrupt climate change. Running NCAR's CCSM3 model, the simulation shows deglaciation during the Bolling-Allerod, Earth's most recent period of natural global warming. Image courtesy J. Daniel (ORNL).

data. Therefore, "productizing" software tools to support rapid deployment at "analysis centers" will likely be a priority. The exact forms these tools will take is not clear. In the past, standalone applications have been the dominant form of distribution. In the future, alternative implementations may be more desirable to better support use in user-customized workflow pipelines running on parallel infrastructure. One potential schematic for implementing these capabilities is shown in Figure 4.

5.1.3 Nuclear Physics

The Nuclear Physics Workshop was held January 26–28, 2009, in Washington, D.C. The recommendations of that workshop are detailed in the report *Scientific Grand Challenges: Forefront Questions in Nuclear Science and the Role of Computing at the Extreme Scale* [74]. It focuses on five major areas in which extreme scale computing is most relevant to nuclear science:

- Nuclear forces and cold quantum chromodynamics (QCD)
- Nuclear structure and nuclear reactions
- Nuclear astrophysics
- Hot and dense QCD
- Accelerator physics

The report notes several key data analysis, visualization, and storage developments that will enable nuclear physics and nuclear astrophysics to advance during the evolution to extreme-scale computing:

- Scientists are concerned about storage and data access mechanisms for the tera-, peta-, and expected exabytes of data from simulations. I/O mechanisms scaling to millions of compute cores, as well as storage infrastructure to scale to the data sizes, will be needed.

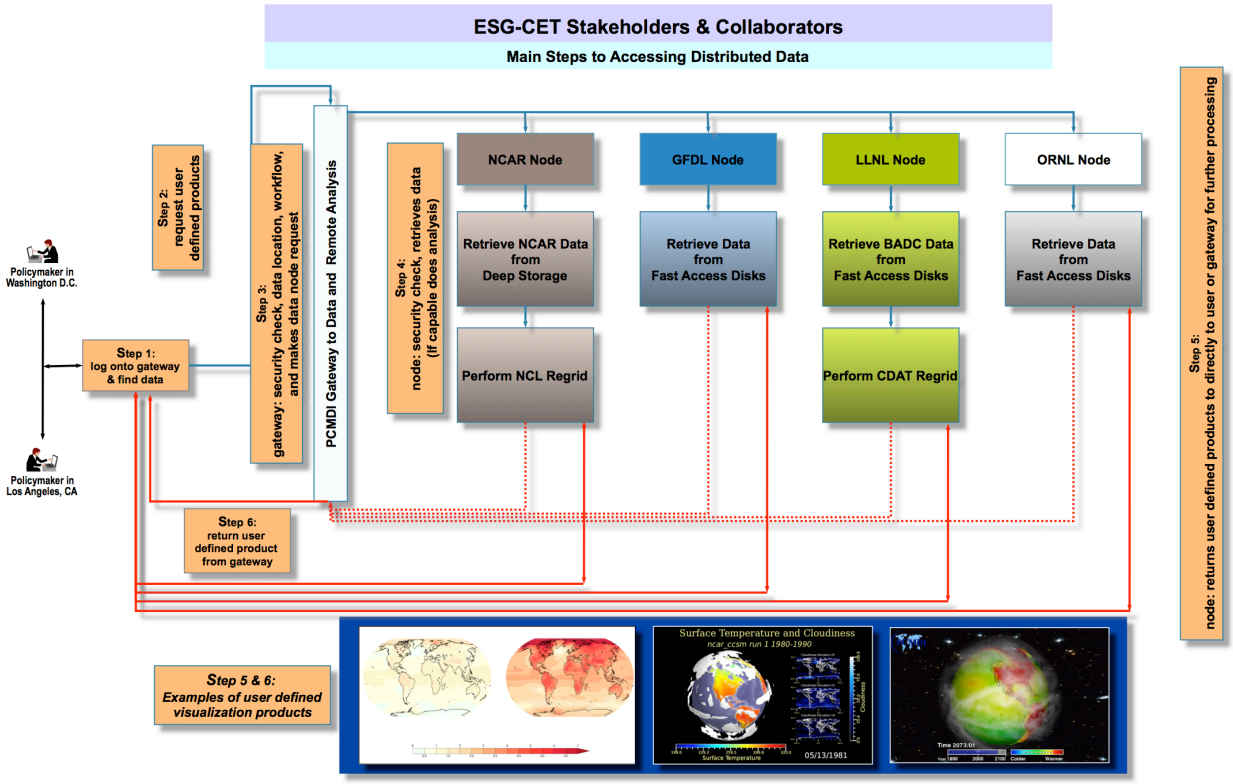

Figure 4: Schematic depiction of a use case supporting remote login for experts (e.g., model developers and climate researchers) and non-experts needing fault-tolerant end-to-end system integration and large data movement. For the expert user, the system expresses all the capabilities and complexities needed for rich data exploration and manipulation. For the non-expert, however, a simple abstraction layer includes easy-to-use controls for maneuvering within the system. Image courtesy of Dean Williams (Lawrence Livermore National Laboratory).

- To contend with the data volumes generated by the simulations, scalable algorithms for data analysis and visualization will be critical.
- The growing volume of data associated with an increasingly ambitious physics program requires sufficient investment in computational resources for post-processing of the data. This will entail the provision of computer systems that are themselves large in scale by current standards, with an aggregate capacity of at least the scale of the extreme (capability) resources themselves. Thus the enterprise of computing will require an "ecosystems" approach to staging, executing, and post-processing data that come from extreme-scale computations.
- Data management approaches for geographically distributed teams are needed.
- Discovery-enabling visualization and analysis of multivariate (scalar, vector, and tensor), multidimensional (as high as six-dimensional), spatio-temporal data must be developed to meet the needs of the science. Comparative analyses between data-sets also are needed.

5.1.4 Fusion

Recommendations from the Fusion Energy Sciences Workshop, held March 18–20, 2009, in Washington, D.C., are detailed in *Fusion Energy Sciences and the Role of Computing at the Extreme Scale* [58]. That report identifies key areas of plasma physics that produce new insights from large-scale computations and considers grand challenges in plasma science to enhance national security and economic competitiveness and

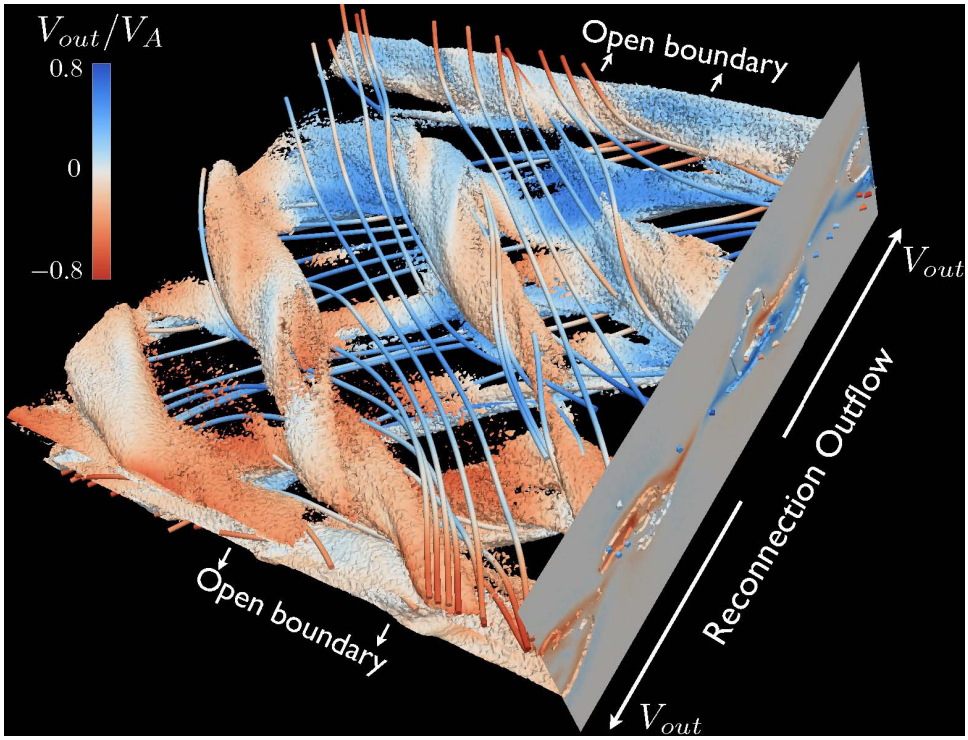

Figure 5: Three-dimensional kinetic simulation of magnetic reconnection in a large-scale electron–positron plasma with a guide field equal to the reconnecting field. This simulation was performed on the Roadrunner supercomputer at Los Alamos National Laboratory using the VPIC code and employing open boundary conditions (Daughton et al. 2006). Shown are density isosurfaces colored by the reconnection outflow velocity. Magnetic islands develop at resonant surfaces across the layer, leading to complex interactions of flux ropes over a range of different angles and spatial scales. Image courtesy of William Daughton (Los Alamos National Laboratory). Figure from the Fusion workshop report [58].

increase our understanding of the universe. The Fusion Energy Sciences workshop featured five key panels: Burning Plasma/ITER Science Challenges, Advanced Physics Integration Challenges, Plasma-Material Interaction Science Challenges, Laser–Plasma Interactions and High-Energy Density Laboratory Physics, and Basic Plasma Science/Magnetic Reconnection Physics (see Figure 5). There were also four ASCR panels: Algorithms for Fusion Energy Sciences at the Extreme Scale; Data Analysis, Management, and Visualization in Fusion Energy Science; Mathematical Formulations; and Programming Models, Frameworks, and Tools.

Participants in the Fusion Energy Sciences workshop identified five key issues in data analysis, management, and visualization:

- Managing large-scale I/O volume and data movement. Techniques need to be developed to optimize I/O performance automatically based on hardware and avoid slowdown due to insufficient rates.
- Real-time monitoring of simulations and run-time metadata generation.
- Data analysis at extreme scale.
- Visualization of very large datasets.
- Experiment–simulation data comparison.

Specific I/O challenges are presented by a few codes and result mainly from PIC techniques, which can output up to 2 PB of data every 10 minutes. This output places severe stress on current technology for in situ processing, as well as for post-processing. Fusion researchers need I/O with low overhead and have been looking into a variety of techniques for reducing I/O overhead and variability. HDF5, NetCDF, and ADIOS

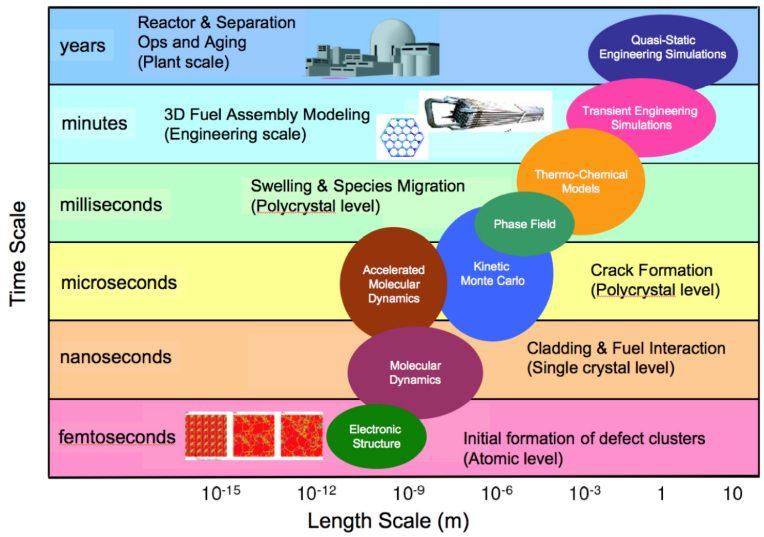

Figure 6: Nuclear energy simulations involve different physical phenomena at varying scales of interest. Figure from the Nuclear Energy workshop report [50].

are the main I/O systems used in fusion science research. Code coupling is also expected to be important for the I/O-like infrastructure.

New techniques that can monitor large-scale simulations easily and allow collaborative, effective monitoring across small groups of researchers during a simulation are needed to enable the science. Advances in data movement are also needed to enable efficient movement of data to a group of researchers analyzing output from the same simulation.

Multi-resolution analysis and visualization must also be deployed for visual debugging and interactive data exploration. Key challenges at the exascale are the capability to see particles to provide insight, techniques for query-based tools to quickly search through data, and methods for remotely visualizing data from a supercomputer on individual laptops. Visualizations must be shared, and techniques for sharing them must be developed for the fusion community.

Another set of challenges is associated with scaling current analysis codes to the extreme scale, given that most of the fusion analysis codes use scripting languages such as IDL or Matlab. The use of programming models such as Global Arrays for extreme-scale computation is recommended in the Fusion Energy Sciences workshop report.

Finally, comparison of simulation data with experimental data must be enabled. Techniques for validation and verification must be developed so that researchers can access, analyze, visualize, and assimilate data between "shots" in near real-time to support decision-making during experiments.

5.1.5 Nuclear Energy

The Nuclear Energy Workshop was held May 11–12, 2009, in Washington, D.C. The recommendations from that workshop are detailed in *Science Based Nuclear Energy Systems Enabled by Advanced Modeling and Simulation at the Extreme Scale* [50], which identifies the key nuclear energy issues that can be impacted by extreme computing:

- Performance issues surrounding integrated nuclear energy systems
- Materials behavior
- Verification, validation, and uncertainty and risk quantification
- Systems integration

The extreme-scale simulations to illuminate these issues will create significantly larger and more complex data sets than those currently considered by the visualization and analysis community. Simulations will consider a variety of time and length scales (see Figure 6), that present significant challenges in exploration,

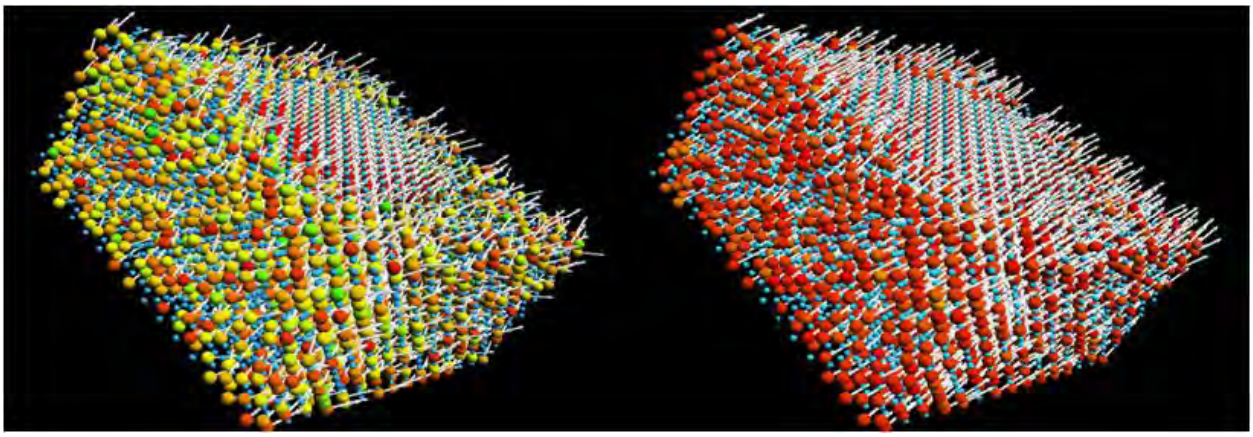

Figure 7: Calculated magnetic spin dynamics in an iron-platinum nanoparticle embedded in a random alloy. Left image is initial state (nonmagnetic); right image is final state (ferromagnetic). Image courtesy of Malcolm Stocks, Aurelian Rusanu, Markus Eisenbach, and Don Nicholson (Oak Ridge National Laboratory); and Yang Wang and Greg Foss (Pittsburgh Supercomputer Center). Image from the Basic Energy Sciences workshop report [24].

detection, and synthesis. The uncertainty quantification effort will lead to ensembles of simulations with as many as one million members, requiring new techniques for understanding the data sets. And very large energy groups, which essentially define a function at every location in the mesh, challenge the existing techniques for multivariate analysis. In terms of scale, a variety of factors will spur increases in data size: improved geometry fidelity (e.g., more complex computational domains), numerical fidelity (e.g., finer resolution and/or higher-order schemes), and physics fidelity (e.g., "more physics"). Finally, although the size of data sets was not explicitly discussed in the report, it should be noted that current thermal hydraulics calculations already contain billions of degrees of freedom per variable per time slice, and that data volume will only increase as more computing power becomes available.

5.1.6 Basic Energy Sciences

Recommendations from the Basic Energy Sciences Workshop, which took place August 13–15, 2009, in Washington, D.C., are detailed in the report *Scientific Grand Challenges—Discovery in Basic Energy Sciences: The Role of Computing at the Extreme Scale* [24]. The report identifies the scientific challenges in basic energy sciences that could be solved or aided by high-performance computing at the extreme scale. The participants identified the following topical areas as the most pressing and critical issues requiring computational modeling and simulation at the extreme scale:

- Excited states and charge transport
- Strongly correlated systems
- Free energy landscapes, rare events, and phase space sampling
- Bridging of time and length scales
- Materials, molecules, and nanostructures by scientific design (see Figure 7);
- Systems and processes out of equilibrium
- Materials properties, including phase diagrams, solvent effects and dielectric properties

The Basic Energy Sciences workshop report stresses a deep concern for the infrastructure that aids in managing and analyzing large data sets. It states that the data challenges will require interdisciplinary developments to design new algorithms, optimization techniques, advanced statistical analysis methods, methods of automated data mining, multidimensional histogramming techniques, data inversion, and image reconstruction. The report notes that data management and analysis infrastructure in use today by the scientific community does not fully exploit new hardware developments such as multicore processors. New and future hardware developments need to be taken into account in addressing these challenges. Data

management and analysis challenges are strongly connected both with the output of simulations and with experimentally collected data. There is also a need for real-time analysis in both cases—simulation and experiment. The report stresses repeatedly the need for dedicated resources for analysis, processing and development, stating that "Major computing systems are not appropriate, nor should they be used, for this type of work."

5.1.7 Biology

The Biology Workshop was held August 17–19, 2009, in Chicago. Its recommendations are detailed in the report "Scientific Grand Challenges: Opportunities in Biology at the Extreme Scale of Computing" [54]. The vision of the grand challenges confronting the biology community over the next few years showcases several different types of data management challenges arising from the extreme volumes of data being produced by simulation, analysis, and experiment. Ranging from the understanding of the fundamental connections between the chemistry of biomolecules and the function of whole organisms, to the processing and interpretation of complex reaction pathways inside organisms used in bioremediation, to the simulation and understanding of the human brain itself, these complex "challenge problems" use data in ways that extend the requirements for extreme-scale data management and visualization.

In particular, the Biology workshop report highlights four distinct areas of grand challenges in the science:

- Understanding the molecular characterization of proteins and biological macromolecules
- Understanding the reaction pathways of cells and the organizational construction of organelles
- Describing and exploring emergent behavior in ecosystems
- Building a computational model of human cognition

In all of these grand challenge areas, data visualization, analysis, and management play a key part; indeed, the final chapter of the report is devoted entirely to that topic. In all of the key biological research areas addressed, there is an important intersection between simulation data and experimental data. Thus analysis and visualization tasks oriented around large data integration and verification are prominent in the requirements. In fact, one of the key concerns highlighted in the report has to do with innovations in high-throughput sequencing equipment; it is not concerned solely with large simulation data volumes. From the perspective of the report, the extreme-scale computing challenges in biology will depend heavily on the ability to adequately support data management, analysis, and visualization.

5.1.8 National Security

The National Security workshop was held on October 6–8, 2009 in Washington, D.C. The report *Scientific Grand Challenges in National Security: The Role of Computing at the Extreme Scale* [41] summarizes the findings and recommendations from this workshop. The participants organized panels in five topic areas, each including some unique needs for data management, analysis, and visualization:

- **Multiphysics Simulation:** Data analysis techniques must span distinct solution spaces in both space and time and must include capturing and analyzing "voluminous data" such as that from sensor networks.
- **Nuclear Physics:** Scales between the nuclear and the atomic must be bridged not just by simulation codes themselves but also by the visualization and analysis tools.
- **Materials Science** specifically stated a broad need for visualization and analysis research and development targeted for exascale simulation data.
- **Chemistry:** Extracting thermophysical properties of localized regions of space requires a higher temporal analysis than is possible using storage and post-analysis techniques.
- **Science of Non-proliferation:** The panel describes this area as a data-driven science. Data extraction and aggregation spans massive, disparate types of data; and machine-assisted analysis is required to aid human analysts in detecting anomalous behavior through sorting, filtering, classification, and visual metaphors.
- **Uncertainty Quantification:** Data management is just as critical for the properties of the computed data as for the computed data itself. Data exploration in this space is particularly confounded by the explosion in dimensionality and the exabytes of data produced.

Several cross-cutting issues were identified among the workshop panels, including uncertainty quantification, image analysis, and data management tools. A number of these areas depend heavily on the ability to organize, analyze, and visualize data; in its conclusion, the report recommends that the national security areas need a "balanced and complete infrastructure" associated with extreme-scale facilities, including "supporting software, data storage, data analysis, visualization, and associated analytical tools."

5.2 Common Themes and Cross-Cutting Issues in Science Application Areas

While the previous section discusses a diverse range of science questions and challenges, there are a number of cross-cutting themes that span the surveyed science areas.

Science applications are impacted by the widening gap between I/O and computational capacity. I/O costs are rising, so, as simulations increase in spatiotemporal resolution and higher-fidelity physics, the need for analysis and visualization to understand results will become more and more acute. If I/O becomes prohibitively costly, it seems likely that the need to perform analysis and visualization while data is still resident in memory will similarly become increasingly acute.

Science applications will be producing more data than ever before. The expanded capacity will result in higher spatiotemporal resolution in computations. Researchers in many areas of science express concern that their traditional approach of writing files for subsequent analysis/visualization will become intractable as data size and complexity increase and the cost of I/O rises. The greater spatiotemporal resolution, combined with increasing complexity and alternative mesh types, puts stress on existing data management, data modeling, and data I/O software and hardware infrastructure. The familiar problems of storing, managing, finding, analyzing/visualizing, sharing, subsetting, and moving data are projected to become increasingly acute as we head into the exascale regime.

Some science applications will leverage increased computational capacity to compute new data types not seen before in visualization and analysis. For example, emerging nuclear transport codes in nuclear physics and astrophysics compute "energy groups," which essentially define a function at every mesh location. The function may be, for example, radiation flux at different frequencies and azimuthal/elevation angles. Such data types, especially for very "wide" energy groups, are outside the scope of most traditional visualization and analysis tools.

Many science applications will increasingly compute ensembles to study "what-if" scenarios, as well as to understand the range of potential behaviors over a range of different conditions. Whereas traditional approaches for visualization and analysis typically facilitate examining one dataset at a time, the growth in ensemble collections of simulations will make it increasingly important to quantify error and uncertainty across the ensemble, as well as between ensembles. Visualizing uncertainty and error is an ongoing research challenge, as is managing increasingly large and diverse ensemble data collections.

Many science areas will need dedicated computational infrastructure to facilitate experimental analysis. One recurring theme is that much scientific activity focuses on experimental analysis, often on dedicated computational infrastructure that is not of the same capacity as the largest supercomputers. This issue is also critical for applications that enable verification and validation leveraging both simulations and experiments. Multiple areas of science have indicated that their experimental analysis is increasing, yet their software infrastructure is incapable of leveraging current, much less emerging, multicore/many-core platforms.

Exposure to transient failures will increase the need for fault-tolerant computing. It is widely accepted that mean time between failure (MTBF) will go down as processor count goes up and system complexity increases. Yet the path forward for having science codes, including visualization and analysis, become fault-tolerant is not entirely clear.

Most existing science applications are written in MPI, which will likely be inadequate at billion-way concurrency. The path forward to creating scalable codes that run at billion-way concurrency and that are portable to different classes of machine architectures is not yet clear. Many groups are investigating hybrid-parallelism using combinations of MPI for distributed-memory parallelism and other approaches, like POSIX threads and/or CUDA/OpenCL, for shared-memory parallelism. Many teams are likely to rely on evolution of fundamental core mathematical infrastructures, like linear algebra solvers, to take advantage of future architectures, rather than undertake performing their own research on portable, high-concurrency solvers.

Automation of science workflows. In order to deal with the massive data sets developed and processed by these new systems, most science areas have a need for end-to-end solutions for DMAV and automation of the workflows associated with the simulations.

6 Research Roadmap

The research roadmap outlines the results of the breakout groups at the exascale DMAV workshop, which explored the driving forces behind exascale analysis needs and possible solutions. They summarize our suggestions for fruitful research directions as the community heads toward exascale. Section 6.1 describes the challenges exascale computing imposes on our current visualization methods, including a push toward significantly increasing processing done concurrently with the simulation. Many of the upcoming changes in computer architecture alter design decisions for visualization algorithms and necessitate a fundamental shift in the data analysis workflow. Section 6.2 addresses visualization gaps formed by changes in user needs and discusses new abstractions required to understand the new forms of data expected by the science domains. Section 6.3 considers fundamental changes in the storage hierarchy and proposes a possible I/O system that is a first-class citizen in any data analysis framework. Finally, Section 6.4 describes the inadequacies of our current data-management tools and proposes new technologies to better organize and use data.

6.1 Data Processing Modes

This section describes the changes that are expected in the methods by which scientific discovery for HPC is performed, that is, changes in *how* simulation data is processed. Possibly more than any other area, the methods by which data is being processed will be affected by the future architectural changes outlined in Section 4. One of the strongest messages from the exascale workshop was that processing data in situ, concurrently with the simulation, will become preeminently important in the next 5–8 years. Of course, traditional post-processing will still form a critical part of any analysis workflow. We discuss both topics in the following section.

6.1.1 In situ processing

Post-processing is currently the dominant processing paradigm for visualization and analysis on ASCR supercomputers (and other supercomputers): simulations write out files, and applications dedicated to visualization and analysis read these files and calculate results. However, supercomputers that have come on line recently are increasing memory and FLOPs more quickly than I/O bandwidth and capacity. In other words, the I/O capability is decreasing relative to the rest of the supercomputer. It will be slow to write data to disk, there will not be enough space to store data, and it will be very slow to read data back in. (See details at the third bullet in Section 4.1.) This trend hampers the traditional post-processing paradigm; it will force simulations to reduce the amount of data they write to disk and force visualization and analysis algorithms to reduce the amount of data they read from disk. Recent research has tried to address the "slow I/O" issue through a variety of techniques, including in situ processing, multi-resolution processing, and data subsetting (e.g., query-driven visualization). However, at the exascale, a second architectural factor will strongly favor in situ processing alone: power limits will discourage moving data between nodes. Although techniques like data subsetting and multi-resolution will still be used, the exascale machine will force them to be applied in situ: data will have to be compressed, discarded, or otherwise processed while it is being generated by the simulation on the large HPC resource.

Benefits of in situ processing. Sparing some supercomputing time to process, structure, reduce, or visualize the data in situ during the simulation offers several benefits. In particular, when data reduction becomes inevitable, only during the simulation time are all relevant data about the simulated field and any embedded geometry readily available at the highest resolution and fidelity for critical decision making. After data reduction, all such relevant data and information would be prohibitively expensive to collect again and compute during a post-processing step. The key aspect of in situ processing is that data are intelligently reduced, analyzed, transformed, and indexed while they are still in memory before being written to disk or transferred over networks. In situ analysis and visualization can extract and preserve the salient features

in the raw data that would be lost as a result of aggressive data reduction, and can characterize the full extent of the data to enable runtime monitoring and even steering of the simulation. In situ data triage can effectively facilitate interactive post-processing visualization.

Barriers to in situ processing. In situ processing has been successfully deployed over the past two decades [28, 31, 38, 75, 71]. However, its use still has not gone "mainstream" for three main reasons:

1. There are software development costs for running in situ. They include costs for instrumenting the simulation and for developing in situ–appropriate analysis routines.

2. There can be significant runtime costs associated with running in situ. Running analysis in situ will consume memory, FLOPs, and/or network bandwidth, all of which are precious to the simulation. These costs must be sufficiently low that the majority of supercomputing time is devoted to simulation.

3. At the exascale, resiliency will be a key issue; in situ analysis software should not create additional failures, and it should be able to perform gracefully when failures occur.

The first category, software development costs, breaks down into two main areas: (1) the costs to couple simulation and analysis routines and (2) the costs to make analysis routines work at extremely high levels of concurrency. These areas are discussed in more depth in the following paragraphs.

It takes considerable effort to couple the parallel simulation code with the analysis code. There are two primary approaches for obtaining the analysis code: writing custom code or using a general purpose package. Writing custom code, of course, entails software development, often complex code that must work at high levels of concurrency. Often, using a general purpose package is also difficult. Staging techniques, in which analysis resources are placed on a separate part of the supercomputer, requires routines for communicating data from the simulation to the analysis software. Co-processing techniques, which place analysis routines directly into the memory space of the simulation code, requires data adapters to convert between the simulation and analysis codes' data models (hopefully in a zero-copy manner) as well as a flexible model for coupling the two programs at runtime.

Further, making analysis routines work at very high levels of concurrency is an extremely difficult task. In situ processing algorithms thus must be at least as scalable as the simulation code on petascale and exascale machines. Although some initial work has been done that studies concurrency levels in the tens of thousands of MPI tasks [45, 15, 30], much work remains. This work is especially critical, because slow performance will directly impact simulation runtime. As an example of the mismatched nature of analysis tasks and simulation tasks, consider the following: the domain decomposition optimized for the simulation is sometimes unsuitable for parallel data analysis and visualization, resulting in the need to replicate data to speed up the visualization calculations. Can this practice continue in the memory-constrained world of in situ processing?

Open research questions with in situ processing. To make situ processing a reality, we must fundamentally rethink the overall scientific discovery process using simulation and determine how best to couple simulation with data analysis. Specifically, we need to answer several key questions and address the corresponding challenges:

- To date, in situ processing has been used primarily for operations that we know to perform a priori. Will this continue to be the case? Will we be able to engage in exploration-oriented activities that have a user "in the loop?" If so, will these exploration-oriented activities occur concurrently with the simulation? Or will we do in situ data reduction that will enable subsequent offline exploration? What types of reductions are appropriate (e.g., compression, feature tracking)?

- How do simulation and visualization calculations best share the same processor, memory space, and domain decomposition to exploit data locality? If sharing is not feasible, how do we reduce the data and ship it to processors dedicated to the visualization calculations?

- What fraction of the supercomputer time should be devoted to in situ data processing/visualization? As in situ visualization becomes a necessity rather than an option, scientists must accept "embedded analysis" as an integral part of the simulation.

- Which data processing tasks and visualization operations are best performed in situ? To what extent does the monitoring scenario stay relevant, and how is monitoring effectively coupled with domain knowledge-driven data reduction? If we have little a priori knowledge about what is interesting or important, how should data reduction be done?
- As we store less raw data to disk, what supplemental information (e.g., uncertainty) should be generated in situ?
- What are the unique requirements of in situ analysis and visualization algorithms? Some visualization and analysis routines are fundamentally memory-heavy, and some are intrinsically compute-heavy. Thus some are not usable for in situ processing. We will need to reformulate these calculations. Furthermore, some analysis requires looking at large windows of time. We may need to develop incremental analysis methods to meet this requirement.
- What similarities can be exploited over multiple simulation projects? Can the DMAV community develop a code base that can be re-used across simulations? Can existing commercial and open-source visualization software tools be directly extended to support in situ visualization at extreme scale?

In situ successes to date. In situ processing is clearly a promising solution for ultrascale simulations. Several preliminary attempts to realize in situ processing in both tightly and loosely coupled fashions have shown promising results and resulted in lessons learned, partially addressing some of the issues mentioned.

There have been some successes, including

- Tightly integrating with the simulation [63, 37] and developing highly scalable visualization algorithms [76, 13]
- Decoupling I/O from the simulation [36], staging data to a second memory location, and enabling other codes to interface to the data
- Conducting data triage and reduction according to scientists' knowledge about the modeled phenomena [67, 68]
- Converting data into a compact intermediate representation, facilitating post-processing visualization [61, 62]
- Adding support for in situ visualization to open-source visualization toolkits such as ParaView [42, 6] and VisIt [71]

Strategies for advancing the state of the art for in situ processing. Further research and experimental study are needed to derive a set of guidelines and usable visualization software components to enable others to adopt the in situ approach for exascale simulation. It is imperative that simulation scientists and visualization researchers begin to work closely together. In fact, this effort must be cross-cutting, also involving experts in applied math, programming models, system software, I/O, and storage to derive an end-to-end solution. It should be a collective effort beyond the DOE community to include the National Aeronautics and Space Administration, the National Science Foundation, the Department of Defense, and international partners. A successful approach will lead to a new visualization and data understanding infrastructure, potentially changing how scientists do their work and accelerating the process of scientific discovery.

Blending in situ processing with data post-processing. In situ processing can generate only data products that were specifically requested when a simulation was launched. If all final data products are generated in situ, and no data is written out for post-processing, serendipitous and exploratory scientific discovery is essentially precluded. There will always be the need, across all scientific disciplines, to pose scientific questions that were not known at the time the simulation was run. Rather than being seen as simply a limitation of in situ processing, this situation can be treated as an opportunity to blend in situ and post-processing modes. In situ processing is well positioned to create intermediate data products that are significantly smaller than raw data sets, and indeed there have been projects that demonstrate success

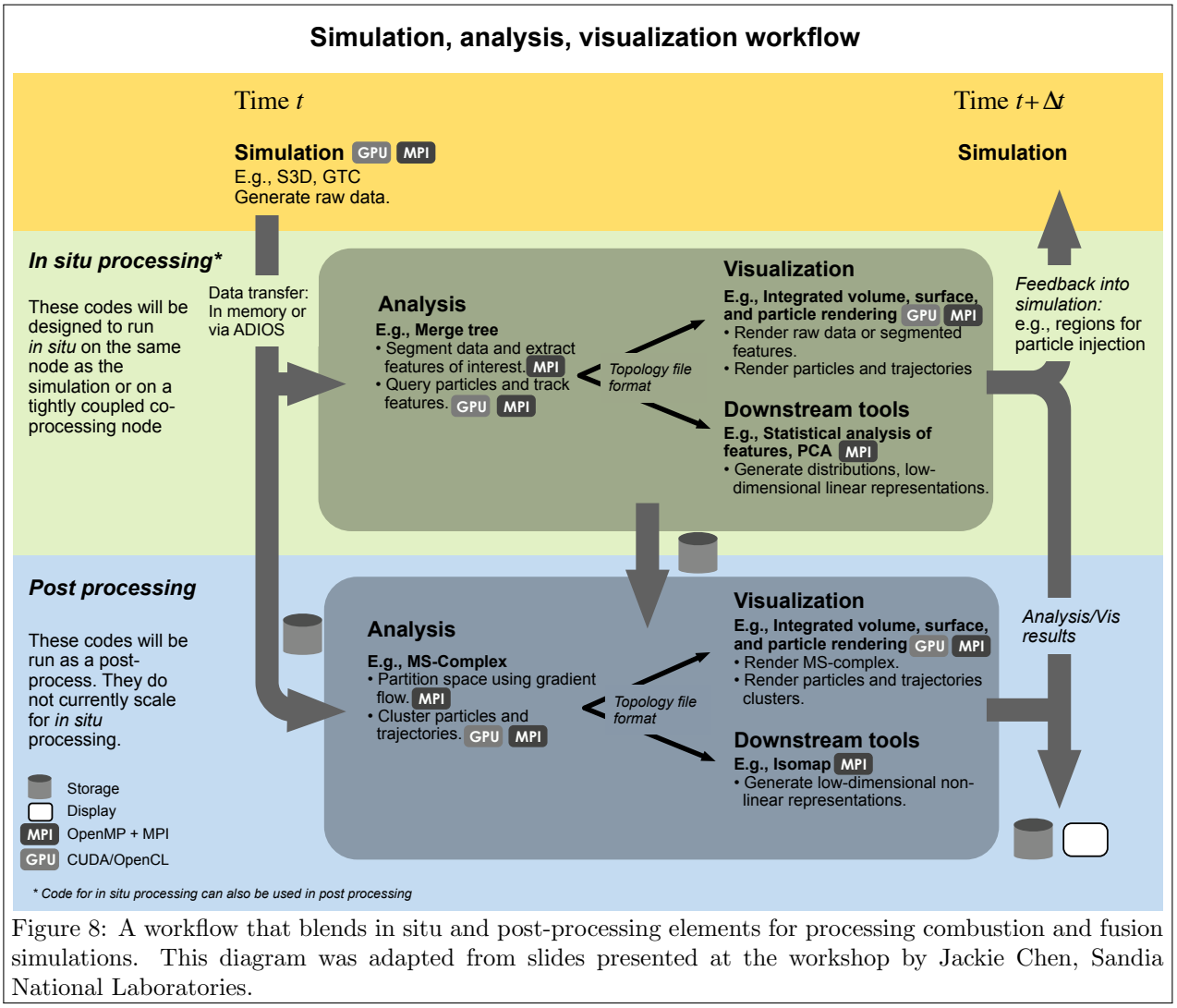

Figure 8: A workflow that blends in situ and post-processing elements for processing combustion and fusion simulations. This diagram was adapted from slides presented at the workshop by Jackie Chen, Sandia National Laboratories.

in doing exactly this. The data post-processing mode can thus use these intermediate products to enable discovery after the fact. Analysis frameworks and algorithms that can be used in either an in situ or a post-processing mode will go a long way toward bridging the gap between the two methods. See Figure 8 for an example of a successful blended workflow.

6.1.2 Data post-processing

Post-processing, or offline, data analysis is the most common approach being applied today in scientific applications. The current widespread use of post-processing data analysis is due in part to its ease of implementation relative to in situ analysis, as well as its ability to facilitate "serendipitous" discoveries in scientific data by freeing scientists from the need to pre-define the type of analysis and the data regions to be analyzed. It can also accommodate any type of data analysis technique, from simple statistics like mean and range to more complex evaluations like a priori rule mining, p-value estimation, or domain-specific preprocessing and analytic kernels. However, the dramatic disparity between FLOPs and memory capacity versus I/O bandwidth expected from future architectures will take conventional post-processing approaches past the breaking point, necessitating a paradigm shift in how offline data analysis is performed.

One consequence of the coming data explosion is that some form of data reduction, such as data sampling and dimension reduction, will be necessary in future scientific applications even before post-processing. Thus

post-processing data techniques must be able to handle the associated uncertainty, necessitating advances in uncertainty quantification.

Future trends in HPC also predict at least three major challenges to offline data analytics. (1) The amount of memory per core is expected to decrease significantly as the number of compute cores increases, forcing future applications and analytics algorithms to be more memory-conscious. Moreover, this problem is not unique to data analysis; e.g., generating checkpoint data often requires additional memory to convert the data for permanent storage [23, 51]. (2) The future power costs of computing are expected to increase dramatically, creating a need for power-aware and energy-conserving approaches. (3) Finally, as the number of processors climbs, there will be a greater and greater need for highly scalable algorithms, especially those that take advantage of emerging new trends in hardware such as solid state drives (SSDs) and NVRAM.

In analyzing the volumes of data produced by the simulations of tomorrow, future analytics approaches will need to extract meaningful information in a scalable and power-efficient manner from ever-larger datasets with proportionally less memory per core. Note that the per-node architectural changes that will bring about the exascale (e.g., greatly increased on-node concurrency, low memory per core, very low I/O per core) will also be seen on local and desktop resources. Thus the methods for local post-processing, even on relatively smaller datasets, are likely to dramatically change. Four nonexclusive strategies for dealing with these challenges and the massive scale of future data are

- Out-of-core analytics
- Approximate analytics
- Index-based analytics
- The use of heterogeneous computing environments

Out-of-core analytics, the use of nonvolatile storage (e.g., hard disk, SSDs, or NVRAM) to store intermediate results, will help offset the reduction in memory per core and improve energy efficiency. The use of approximate techniques and index-based methods for data analysis can reduce the computational and energy cost. And taking advantage of emerging heterogeneous architectures like GPGPUs (general-purpose graphics processing units) can improve scalability and energy use.

Out-of-core data analytics. Out-of-core algorithms save the intermediate values of a computation to secondary storage to conserve the relatively scarce memory resources of the system. Because of the significant and widening gap between memory and I/O performance, such approaches are typically bound by the performance of the institutional I/O infrastructure and have had little success. Worse yet, large parallel systems typically use a single file system that is shared among all of the users of the systems, creating contention and significant variation in performance. Fortunately, emerging storage technologies like SSDs (see Section 4.2) have the potential for low-power, low-latency, and high-capacity parallel I/O within a local machine. As these types of devices are integrated into nodes in high-performance machines, out-of-core analysis is likely to become more practical and effective.

Approximate analytics. The use of approximate analytics has the potential to reduce the complexity of data mining techniques by several orders of magnitude, resulting in substantial energy savings and the capability to handle substantially larger amounts of data.

In many cases, existing data analytics approaches are already approximate. Tasks like data clustering, rule mining, and predictive model fitting all provide results relative to some error, whether by heuristically minimizing a clustering measure, maximizing rule confidence, or minimizing training error. Additionally, data in many domains are subject to inherent noise and uncertainty owing to the ways the data was collected or the models used to process the data. In the biological domain, for example, protein interaction data may contain a large number of spurious (false positive) interactions [66, 52]. And there are many cases where exact results are not necessary. Finally, approximate algorithms can provide a cheap way (in terms of compute time and system resources) to evaluate and fine-tune parameters or to identify areas of interest that merit further exploration. The use of a coarse-grained visualization of the data space to guide data analysis is known as *visualization-guided* analytics.

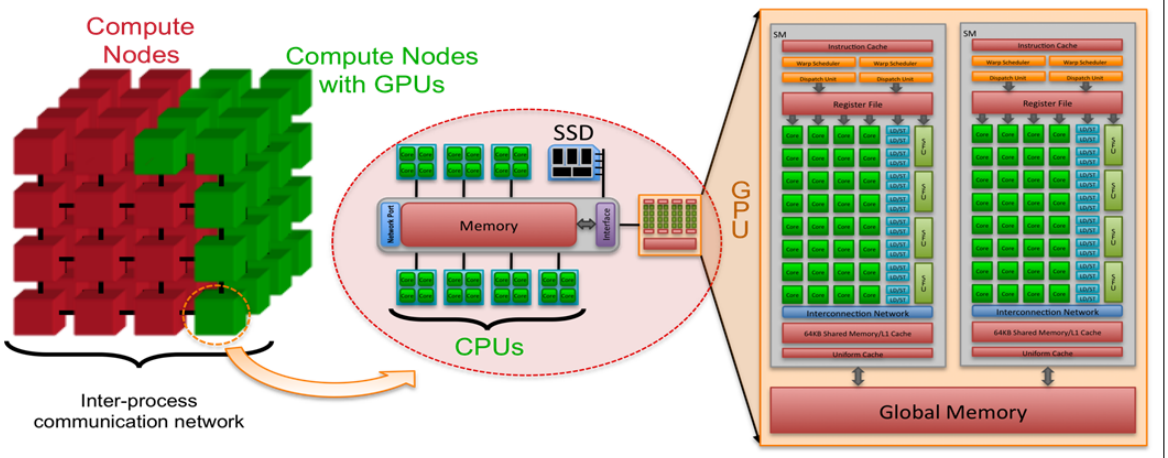

Figure 9: Illustration of machine with heterogeneous architecture. Each node has multiple CPU cores, and some of the nodes are equipped with additional computational accelerators, such as GPUs.

Index-based computation. Another important tool for improving the performance and energy-efficiency of exploratory analytics techniques, the use of data indices, can drastically reduce the amount of computation required for data selection (e.g., identifying all variable values that fall within a specified range) or calculating simple statistics such as mean, median, maximum, and minimum. Pre-generating simple histograms using indexing has also been shown to be I/O-efficient and conducive to rapid scientific discovery [73]. Indexing may also be incorporated into various data mining techniques to improve performance. For example, a subtree in a decision tree can be viewed as a clustering of records satisfying the decisions applied to the root of the subtree, and using data indices to retrieve the necessary values can reduce overall computation time. Indices may also be used in computations involving much redundant computation (e.g., all-pairs similarity search [3]), resulting in large computational and energy savings.

Heterogeneous architectures. Given the massive scale of future data analysis applications, future approaches must take advantage of the acceleration offered by alternative hardware architectures (e.g., GPGPUs, FPGAs, accelerators). These specialized architectures offer various programmatic challenges, but they can achieve significant time savings over traditional general-purpose CPUs for algorithms adapted to their specific requirements. For example, applying GPGPUs to analytic algorithms with strong data independence can speed up calculation by more than an order of magnitude compared with state-of-the-art CPU-based algorithms [2]. Figure 9 illustrates a potential future heterogeneous high-performance system equipped with GPUs and SSDs.

6.2 Data Abstractions

One of the clear take-away messages from the Scientific Grand Challenges Workshop Series reports is that the dramatic increases in computational power to come as we approach the exascale will allow exploration of scientific disciplines and physics that have not yet been practical to explore. As one example, the next several years of HPC climate simulation will involve the addition of sea ice to global climate models and the simulation of the full carbon, methane, and nitrogen cycles. These new areas of scientific exploration are all likely to require new fundamental visualization and analysis techniques. This section discusses some of the new *data abstraction* methods required for understanding new scientific phenomena. In addition, we review software engineering and infrastructure changes that will be necessary to deploy these new methods of understanding.

A major challenge in integrated DMAV is the need for coherent development of a large class of algorithmic techniques that satisfy the constraints of the computing environment described in the previous sections, including, for example, the ability to exploit computing resources in both in situ and post-processing modes.

Techniques can be classified in terms of the class of *data abstractions and exploration modes* that they provide to aid scientists in extracting knowledge from large, complex datasets derived from experiments and from petascale and upcoming exascale simulations. Fundamental advances are needed in these areas to ensure that bottlenecks in this infrastructure do not invalidate the major investments made to accelerate the scientific discovery process.

6.2.1 Extreme Concurrency

Existing visualization and analysis tools are primarily geared toward CPU-based architectures. Many of these analysis tools, including ParaView [53] and VisIt [33], are also adept at effectively using distributed memory parallel machines and can scale to current leadership computing platforms [16]. However, with the proliferation of hybrid architectures, effective ways to fully exploit these new computers are critical. See "Heterogeneous architectures" in Section 6.1.2.

Existing visualization algorithms need to be redesigned to make efficient use of multi-core CPUs and GPU accelerators. Scalable implementations of visualization and analysis software to fully exploit the diverse future systems—including hybrid cores, nonuniform memory, various memory hierarchies, and billion-way parallelism—will be needed to meet the needs of the science. Effective programming models must be leveraged toward this effort.

6.2.2 Support for Data Processing Modes

In situ processing is a necessary technology for data processing, as discussed in Section 6.1.1. The technology is also important for building and using data abstractions. Directly embedding data analysis and visualization within simulations will enable them to interact with simulations as they run, to monitor and understand their progress, as well as help in setting up new simulations. Given the growing complexity of computer systems, visualization and analysis will be key to helping debug simulations and identify performance bottlenecks. Both the fusion and the basic energy sciences communities (Sections 5.1.4 and 5.1.6, respectively) specifically note the need for real-time monitoring and analysis of simulations.

Another key investment area is a co-scheduling mechanism to couple visualization resources with computation resources. Since not all needs can be met by running visualization and analysis in the same program space as simulation, co-processing is a likely alternative that retains many direct coupling characteristics, including full access to all simulation data. This was identified as a key research area in the Scientific Data Management breakout group (see Section 6.4). A likely solution for co-scheduling is the use of I/O middleware as described in Section 6.3.2. Both high-energy physics and fusion (Sections 5.1.1 and 5.1.4, respectively) call for an I/O paradigm for analysis and code coupling.

Although in situ analysis is essential for exascale computing, post-processing is expected to remain a fundamental exploratory mode. (See Section 6.1.2 for a complete treatment.) When data abstractions are generated in post-processing mode, the I/O time required to read data is critical. Data generated by simulations may be in a format optimized for minimizing the I/O time of the simulation and may not be optimal for future analysis [46]. These formats are typically not optimized for reading by visualization and analysis applications. Furthermore, little if any metadata is added to help support the analysis. Generating and serving data abstractions efficiently and interactively requires data formats that are friendly to visualization and analysis, with efficient layout for I/O, and multi-resolution and hierarchical layouts that allow for better navigation on scales from global to local [1]. Therefore, data formats are a key technology for exascale. They are highlighted by the climate and nuclear physics communities (Sections 5.1.2 and 5.1.3, respectively), and scientific data formats are of great interest to DMAV in general (see Section 6.3.3 for a detailed discussion.)

6.2.3 Topological Methods

A new class of data analysis techniques based on topological constructs has become popular in recent years because of their proven ability to extract features of interest for science applications [32, 27, 18, 9, 10, 39] and their basis in robust combinatorial constructs that make them insensitive to numerical approximation and noise in the data. These topological methods can achieve massive data reduction while representing complete families of feature spaces that scientists can still explore in the knowledge extraction process. Providing wide access to these methods by exascale simulations will yield major advantages for scientists, but fundamental

research is still needed to achieve the needed level of maturity. Specific investment is needed in their approximation and parallelization, since they are usually based on constructs requiring global propagation of information, especially if computed exactly. Multiscale representations will also have a central role, since they are primary enablers of orders-of-magnitude improvements in data reduction with explicit control of the quality of the information preserved. Research is also needed to generalize discrete methods that have been successful in Morse theoretical frameworks [21, 22] so they can be applied to more general vector field representation and analysis. This long-term investment in mathematical and and computer science research should include the development of new standardized file formats for the compact representation of feature spaces that can be computed in situ and that users can explore in real time or in post-processing.

6.2.4 Statistical Methods

Simulations and data harvesting instruments continue to increase their spatiotemporal resolution and continue to add new physics components, resulting in higher data density. This increased data density runs counter to decreasing memory-per-compute-cycle ratio trends discussed in Section 4. The old mantra of mathematical statistics to extract maximum information (read "use maximum cycles") from minimal data (read "use minimal memory") comes back as we go toward exascale. Data economy becomes synonymous with power economy. Here, maximum information from minimal data is still the relevant optimization problem, but its context is different. Data are now highly distributed and sampling of data runs counter to efficient cache utilization. Optimal sampling theory of mathematical statistics is highly developed and includes rigorous uncertainty quantification but it too needs further research and tool development for this new context. Fast progress requires interdisciplinary teams that understand mathematical statistics as well as future computer architectures.

Statistical analytics are aimed toward gaining insight into a postulated data generation process (often termed the "statistical model" or the "likelihood"). Usually, the entire spatial and time extents of the data are used to estimate the parameters and uncertainties of the data generation process. This is true of both frequentist and Bayesian methods. Bringing these important methods and their uncertainty quantification to the exascale requires careful redesign of the underlying estimation algorithms toward a single pass through the data. While completely novel approaches are possible they will be evaluated with respect to current methods, which possess data optimality properties that lead to their original design. Progress here requires a complete redesign of the underlying mathematics with a clear understanding of target computer architectures.

6.2.5 Support for Large Distributed Teams

Grand challenges in modern simulation and experimental science are increasingly tackled by large teams of people. To solve complex multiscale, multiphysics, multimodel challenges, simulation teams consist of individuals from diverse backgrounds. They are geographically distributed, often with collaborating institutions in different countries or continents. This trend is increasing, as massive computing and storage resources cannot be replicated in too many locations, and the network infrastructure is growing in capacity and performance even as the amount of data produced is exploding.

Although many analysis tools, including VisIt and ParaView, are capable of bridging distance via remote client/server architectures, and are often used for this purpose, it is also commonplace to transfer large amounts of results data for analysis, visualization, and archiving. Remote visualization is a well studied problem, but we can identify several potential problems at exascale. First, supercomputing facilities typically employ batch-based job submission, and at exascale we expect visualization jobs must share these facilities [14]. However, interactive visualization jobs must be launched on interactive queues so that they will start while the user is available. Interactive queues require nodes to remain idle so that jobs can be created immediately. This is contrary to traditional queuing policies that keep the computer as busy as possible.

Another issue is that analysis and visualization codes, which already prefer computers with "fat memory" to support their data-intensive operations, have not been designed to be memory-conservative. For example, a ParaView executable itself can take up to 200 MB of memory. As computing moves to future architectures characterized by their low memory footprint per core, the architecture of such tools needs to be modified to use as little memory as possible.

Because they are usually run for short time scales, data analysis and visualization tools are also generally poor at fault tolerance. With system faults expected to be more common at exascale, visualization and analysis tools will need to incorporate fault-tolerant algorithms as well as implementations. No existing visualization toolkit supports fault tolerance, and investment in this effort is required.

Remote visualization must extend beyond communication between a single data host and single user, as science is increasingly collaborative and teams are distributed worldwide. The climate community (Section 5.1.2) in particular calls for a consortia of tools and extreme-scale data serviced over a wide area network. The nuclear physics and fusion communities (Sections 5.1.3 and 5.1.4, respectively) also require data management and collaboration among geographically distributed teams. An infrastructure that enables scientists to collaborate effectively to solve problems will be critical in the exascale computing era. Data streaming mechanisms and multi-resolution techniques will also be essential for exploring remote data. Fast access to progressively refined results, with clear understanding of error bounds, will be critical to the remote visualization infrastructures to be deployed.

6.2.6 Data Complexity

Many factors lead to an increased data complexity to be managed at exascale. Most of the science application drivers reviewed in Section 5.1 cite the need for multiscale, multiphysics, and time-varying simulations. Effective visualization and analysis algorithms of these results are necessary to meet the needs of the science. Investment will be needed to incorporate novel techniques, including machine learning, statistical analysis, feature identification and tracking, and data dimensionality reduction, all to make the data more tractable.

Moving forward in response to the complexity of the science being studied, it will be common for different simulation codes to operate at different scales, both physical and temporally. These coupled codes will provide input to one another during the simulation process and probably will also produce independent results that need to be visualized at the same time for maximum insight. Existing tools do not have the needed infrastructure to do so in a straightforward manner. Even more, the existing systems do not provide feedback to the end user about uncertainties introduced in the mapping.

Scientists already recognize several limitations of current data abstractions. Fusion (Section 5.1.4) requires better methods to visually analyze particles and better query-based tools to quickly search through data. Nuclear energy (Section 5.1.5) notes that current multivariate techniques are not satisfactory for visualizing energy groups, which define a function at every location in a mesh. Basic energy sciences (Section 5.1.6) calls for a variety of new techniques, including advanced statistical methods, automated data mining, multidimensional histograms, data inversion, and image reconstruction.

As science becomes increasingly multi-domain in nature, multiphysics and multiple scales lead to more complex data structures. Hierarchies are becoming finer-grained, with more applications refining smaller groups of elements—sometimes even refining elements individually. High-order fields have become more common outside of finite element codes, and high dimensionality is used more frequently as physics algorithms become more sophisticated. Increasingly expressive data structures are being generated that use graphs, hierarchies, and blocks with data mapped to various topological elements and overlaid meshes. Additionally, scientists need to be able to understand time-varying phenomena, which requires the capability to stride over these complex datasets. To effectively draw insight from these complex data, new techniques are needed, including tensors, topology, information theory, and data mining.

6.2.7 Comparative Visualization

Analyzing data from extreme-scale computing requires advanced comparative techniques. For example, science applications like fusion (Section 5.1.4) require experiment-to-simulation data comparison. Comparisons among simulations also will become commonplace. Increases in the availability of computing cycles do not always lead to increasing dataset sizes. Rather, increased computing cycles can also be used to run the same simulation multiple times, thereby generating a number of datasets. Infrastructure continues to be limited in providing for ways to quantitatively compare the results of different simulations.

6.2.8 Uncertainty Quantification

Uncertainty quantification is a technique for measuring the envelope of potential physical responses to an envelope of possible input conditions by repetitively running a simulation over samples of the possible input conditions. The most prevalent example of this in the last 10 years is climate simulation (Section 5.1.2). Nuclear energy research (Section 5.1.5) will require uncertainty quantification using ensembles with as many as one million members. However, visualization of uncertainty quantification has been a research topic for over 10 years, and tools have yet to be deployed. Although uncertainty quantification is used increasingly in multiple scientific domains, methods for visualizing ensembles [49], as well as uncertainty in general [48], are still in their infancy. Furthermore, the uncertainty needs to be appropriately handled in the various components involved, including the physics employed and the math solvers used.

6.3 I/O and storage systems

As computing power in the DOE facilities continues to grow, I/O continues to challenge applications and effectively limit the amount of science that can be performed on the largest machines. Reducing the I/O impact on calculations is a major challenge for I/O researchers. It is generally difficult to achieve good levels of performance for both restart data and analysis/visualization data on large- scale parallel file systems. I/O performance often limits application scientists, reducing the total amount of data they write, and they often make ad hoc decisions about where and when to write out data.

As exascale computing approaches, it is useful to review lessons learned from the DOE computing facilities and consider research conducted to enhance our knowledge about how to create fast I/O systems and link visualization, analysis, and data reduction in an easy-to-use system. Research has shown it is possible to create self-describing file formats that have simple APIs (e.g., HDF5 [60], Parallel NetCDF [64], ADIOS [35]), and use I/O middleware such as MPI-IO, commonly the backbone of many higher-level I/O systems, to enable high-performance application I/O. Furthermore, data needs to be indexed appropriately so that complex queries can be performed in a reasonable amount of time. The following metrics are a guide, and they need to be carefully evaluated for future research:

- User adoption
- Read/write I/O performance, compared with the peak I/O performance of the file system
- Resiliency
- The overall energy used for I/O

This section discusses current and future storage systems, I/O middleware to bridge the gap between storage system and application, scientific data formats for storage and transit, ways to link with application data models, and database technologies that should be incorporated into these software layers.

6.3.1 Storage Technologies for the Exascale

A key challenge for exascale computing is efficient organization and management of the large data volumes used and produced by scientific simulations and analyzed and visualized to support scientific investigation. The challenge exists at three levels:

1. At the architecture or node level, to efficiently use increasingly deep memory hierarchies coupled with new memory properties such as the persistence properties offered by NVRAM
2. At the interconnect level, to cope with I/O rates and volumes that can severely limit application performance and/or consume unsustainable levels of power
3. At the exascale machine level, where there are immense aggregate I/O needs with potentially uneven loads placed on underlying resources, resulting in data hot spots, interconnect congestion, and similar issues

For future I/O systems at the exascale, we envision an environment in which, at node level, there are multiple memory sockets; multiple coherence domains (i.e., “clusters” on a chip); multiple types of memory, including NVRAM; and disaggregation in which certain noncoherent memory is disaggregated and reachable via on-chip PCI or similar interconnects (see Section 4.1 for more details). Caps on power resources will

result in tradeoffs between data movement on-chip and on-node versus movement to where analysis and visualization computation is applied to the data. The result will be tradeoffs in data movement on-chip and on-node versus movement to where analysis and visualization computation is applied to the data. Analysis code can be moved to where data currently resides (in situ processing, see Section 6.1.1), but this may introduce substantial variations in code execution times that degrade the speedup of parallel codes. A more power-efficient solution may be moving select data to other on-node memory, including persistent memory, before performing analysis. This may also be necessary to retain checkpoint state in case of failure and may even require multiple copy operations (e.g., also moving data onto disaggregated memory blades) to guard against node failure. In such cases, there is both a necessity and an opportunity to apply operations to data as it is being moved. In summary, even on-chip and on-node, there will be tradeoffs in where and when analysis or visualization can or should be performed. For the I/O system, the overwhelming requirement is flexibility in *which* operations are applied *where* on-node and *when* they are applied—whether synchronously with the application, thus potentially affecting its execution time, versus asynchronously and during on-node data movement.

Beyond individual nodes, operations that require use of the interconnect are expensive in terms of both performance and power. As a result, it may be preferable, on both counts, to perform analysis by moving data to a smaller number of nodes —in situ data staging—prior to analysis. Data staging also offers opportunities for hardware differentiation by endowing staging nodes with substantially more memory, large NVRAM, and extensive flash-based storage. For I/O systems, the result is a need for multi-tier solutions to I/O that will allow analysis or visualization tasks to be associated with I/O on-core, on-node, and in-staging, and allow in-staging analyses, because they are inherently more asynchronous, to be substantially more complex than those on-node. That is, in-staging computation, especially for well-endowed staging nodes, probably can operate on multiple output steps and will be bound by timing requirements determined by output frequencies.

The final memory tier considered for I/O is large-scale disk storage. We expect that, as has been the case in HPC for decades, storage will be attached to the periphery of the exascale machine or to ancillary network-connected machines. It is likely that its throughput levels will be far below the aggregate I/O throughput level of which the machine's cores are capable. As a result, a paramount requirement for data analysis and visualization is "to bring code to data" and to do so dynamically when (i.e., at which output steps or simulation iterations) and where (i.e., at which parts of the simulation, and thus at which cores and nodes) certain analyses or visualizations become important.

6.3.2 I/O Middleware

One of the main accomplishments for parallel I/O has been the inclusion of MPI I/O [59] in MPI distributions, including MPICH2 and OpenMPI. This has enabled middleware developers to layer their software (e.g., Parallel NetCDF, HDF5, ADIOS) on top of this layer, giving application developers a simple stack to generate their I/O request (MPI-I/O or POSIX writes) or use the higher-level middleware directly. As research has shown, there are many challenges to using this multi-tier layered approach. Because of the imbalance between compute power and I/O bandwidth in computer systems, petascale computing requires I/O staging, which deals with the movement of data from the compute nodes to another set of nodes that can process data "in situ" and then write data out. Data movement (synchronous versus asynchronous) and the operations that can run on a staging resource are active areas of research that will be critical to effectively scaling I/O performance to exascale.

As I/O infrastructure is a shared resource, performance degradation due to other concurrent I/O jobs is a growing concern. As we move closer to the exascale, diverse networks that use different forms of remote data memory access (RDMA) and multiple tiers of storage (NVRAM) further complicate the matter. Another complication is the increased complexity of hundreds of cores on a node, which allows staging resources and the computation to run on cores on the same node. If these complications are not managed and scheduled properly, they can increase the jitter in the system significantly. Mechanisms will be needed to reduce the variability of I/O on the system, using techniques like Quality of Service. I/O middleware will need to exploit the diverse network topology of future interconnects to ensure that network pathways are used effectively for data movement. Additionally, even within a node, code developers will need tools to leverage the parallel data paths provided by multi-lane network interface cards and features including RDMA. As the number of cores in a system and the number of researchers accessing the file system grow, new techniques to adaptively

write out data are needed. File formats, typically optimized for write workloads, should also support efficient read-access for analysis and visualization. There is a critical need for research into support for data models of simulations, as well as analysis mechanisms in file formats, efficient layout of data on storage, reduced collectives for I/O, and on-the-fly transformation of data being written to storage into formats suited to post-processing (see Section 6.3.3). Data reduction and analysis techniques in I/O middleware will be critical to reduce the amount of data written to storage and provide faster insight into simulation results. These mechanisms will need to be embedded intelligently, considering the system, simulation, and analysis characteristics, as well as data locality.

6.3.3 Scientific Data Formats

There are several self-describing file formats. The most common formats in HPC are HDF5, NetCDF, and ADIOS-BP (a relatively recent entry). All three of these are metadata-rich and have been used in many communities within DOE for large simulations.

HDF5 has a versatile data model that can represent complex data objects and a wide variety of metadata. Like all of these formats, it is portable across all architectures and has no effective limit on the number or size of data objects in the file. The output creates a hierarchical filesystem-like structure in the file created. HDF5 also contains APIs to write and extract subsets of variables written to the file, both sequentially and in parallel.

NetCDF encompasses three variants: NetCDF3, NetCDF4, and Parallel NetCDF, all self-describing. Arrays can be rectangular and be stored in a simple, regular fashion to allow the user to write and read a subset of variables. NetCDF4, based on HDF5, is fully parallel and can read NetCDF3 files. Parallel NetCDF is a parallel implementation of the NetCDF3 file format.

ADIOS is an I/O componentization that provides an easy-to-use programming interface and is meant to be as simple to use as simple Fortran or C file I/O statements. ADIOS writes to NetCDF4, HDF5, ASCII, binary, and ADIOS-BP file formats. It allows a user to view I/O in a manner similar to streams and to operate on data (in memory) across multiple nodes as well as in files. ADIOS-BP is similar to both NetCDF and HDF5 in that it is portable, is parallel, and allows a user to write and extract subsets of variables. ADIOS also includes automatic generation of statistics and includes redundancy of metadata for performance and resiliency.

Many gaps are emerging in the scientific data format technology being developed on the next-generation systems in DOE computing facilities. This section enumerates the gaps and describes current solutions in an attempt to minimize problems through research into scientific data formats:

1. I/O variability on simulations running at scale. There have been several attempts to reduce variability, but the problem persists and is growing.

2. I/O performance from codes running at scale writing a small amount of data. Most of the current technology solutions show I/O degradation when writing a small amount of data on a large number of processors. As the concurrency increases on each node, the limited memory bandwidth will present new challenges.

3. Read performance for visualization and analysis codes. Since datasets are growing and processors are getting faster, the major bottleneck for most analysis and visualization will increasingly be I/O.

4. Data model mismatch. The data models used by simulation and analysis codes may not match the array-oriented data models implemented in scientific data formats. The mismatch may result in inefficient transformation of data in moving between data models.

One of the most important pieces of future research will be scaling storage systems as datasets continue to grow. It is difficult to imagine efficiently reading and writing a single file that has grown to 1 PB. Pieces of data can become distributed, and the DMAV community must learn to move the "correct" data closer to the machine that will analyze it. Scientific file formats must be flexible, portable, self-describing, and amenable to changes by the community. The file format must allow for links into other files, additional metadata, possible redundant data with a limited lifetime, provenance, and statistics that allow additional information to be included (so long as it does not significantly extend the size of the dataset). The file format must also

be elastic—able to change itself when moved so that it can self-optimize on different file systems to achieve high levels of concurrency and performance. And it must be able to cope with the many failures that can arise in the exascale.

Looking even further ahead, the concept of a "file" may have to be revisited altogether. Scientific data may be stored in some alternative data container, be it a database, object storage, or some other alternative. Fortunately, the specific details of how bytes end up stored persistently can be hidden by the scientific data libraries. Research to ensure that these abstraction layers still deliver high performance is required.

6.3.4 Database Technologies

Database systems are widely used in commercial applications. Traditional database systems are designed for managing banking records and other types of transactional data. Users are presented with a logical view of the data records as a set of tuples and given the Structured Query Language (SQL) for interacting with the data [43]. This simple logical view and its associated data access SQL allow users to work with their data without any knowledge about the physical organizations of the actual data records.

In scientific applications, the usage of databases is primarily limited to metadata, whereas most scientific data are stored as files in a parallel file systems. However, databases are also used in a more substantial way in a few cases. For example, a significant amount of protein and DNA data is stored in centralized databases [65, 5]. In these cases, the volume of data is still relatively small; the bulk of raw DNA sequencing data and spectroscope data are not stored in the database systems. The only known example of a large amount of scientific data managed by a database system is the Sloan Digital Sky Survey (SDSS). Through extensive collaboration with a Microsoft SQL server team, groups of astrophysicists are able to use SDSS to conduct extensive data analysis tasks [26, 25, 57]. A number of characteristics make SDSS data well suited for a database system. For instance, the number of records is relatively modest, there is a widely distributed community of users, and many analyses can be done with a relatively small number of selected data records. Other data-intensive applications may have much larger datasets, and their analyses often require a significant fraction of the data and more complex access to the data.

ROOT is another specialized scientific database engine designed for high-energy physics [12] to manage data produced from experiments such as those conducted on the Large Hadron Collider. ROOT uses an object data model and does not support SQL. It uses an interpreted C++ as its programming language. The wide use of ROOT can be taken as confirmation that SQL is not a sufficiently flexible interface for scientific applications. Another well-regarded method of constructing a scientific database is an alternative data model known as the array model [11], which can be seen as an extension of the vertical databases [8, 56]. A key insight behind this design is that most scientific data can be expressed as arrays, and many of the popular scientific data formats indeed are array based [60, 64, 47]. The basic operations in the new database, named SciDB, will be on arrays. The designers of this system are expanding SQL to support data analyses.

The three database technologies that are most relevant to the scientific applications are data warehousing [72], distributed databases [44], and NoSQL[1].

Typically, a data warehouse has many more data records than a transactional database and demands more advanced acceleration techniques to answer queries efficiently. Many of these acceleration techniques, such as compression and indexing, can be applied directly to scientific data analysis without using a database or data warehousing system. However, the users have to manage these auxiliary data structures explicitly or through a library.

In distributed and parallel databases, a large portion of the data processing is done close to the disks; the distributed file systems can only ship bytes to the clients and cannot do any further processing. This inability to perform computation is primarily due to a limitation in the data model adapted by the file systems, wherein the content of the data is viewed as a sequence of bytes. Most parallel and distributed database systems are purely software based and can be distributed on any commodity hardware. In this case, the parallel data placement, resilience, replication, and load-balancing techniques used by these database systems can be leveraged in the design of parallel file systems at exascale. Furthermore, there are a number of specialized parallel database systems based on specialized hardware, such as Netezza [17] and Teradata [4]. The special feature of Netezza is that it utilizes a custom disk controller to perform part of the database operations; the special feature of Teradata is a unique interconnect network called BYNET.

[1] A good source of information is `http://nosql-database.org/`.

As data volume increases, the performance parallel database systems become unacceptable for a number of applications. The strategy of developing a data processing system with minimal SQL support is taking on considerable momentum; a movement known as NoSQL has emerged recently. A prime example of such a system is Hadoop, based on the MapReduce concept [19, 70]. The original motivation for developing the MapReduce system was to produce an index for all Web pages on the Internet. For data on this scale, the key operation on the database system is a simple filtering operation, such as locating a number of records satisfying a handful of conditions. These NoSQL systems concentrate on performing a handful of simple but essential filtering operations efficiently. By limiting the scope of operations supported and taking full advantage of a large number of commodity computers, NoSQL systems can complete their operations on massive amounts of data in an efficient and fault-tolerant way and at the same time keep the systems easy to use.

6.4 Scientific Data Management

Data management is an extremely broad area, covering many research communities. This section identifies and discusses research areas considered relevant to the ASCR portfolio. The following relevant topics are considered:

- Workflow systems
- Metadata generation, capture, and evaluation, including standards, provenance, and semantics
- Data models, representations, and formats
- Data movement and transfer within a single machine and across machines
- Data fusion and integration
- Data management support for analysis and visualization through appropriate interfaces
- Data reuse, archiving, and persistence, including data integrity and the ability to annotate data and retain the annotations
- Data quality metrics, including tools for statistical measures of data quality and uncertainty
- The ability to quickly identify relevant subsets of data through indexing, reordering, or transforming the original data set
- The ability to publish and find data sets across scientific communities

The workshop panel restricted its discussions to those aspects that fall within ASCR's mission, particularly the impact of the emerging massively multi-core computing paradigm and the associated transition to in situ analysis. Within these focused topical areas, there are some topics for which ASCR is driving the research agenda for the community (e.g., scientific data models and data movement) and others in which ASCR is not actively involved (e.g., publishing scientific data sets).

Although some may argue that improving data management technology is an end unto itself, within this community it is seen as a means to the ultimate goal of improving scientific discovery. Data management is expected to become the critical bottleneck for improving computational science productivity within the next decade. This is a result of the disruptive change expected as power consumption limits improvements in computational resources as exascale computing approaches. As has been noted in previous reports, providing just the storage and networking bandwidth traditionally available to and desired by scientists would significantly exceed the budget for an exascale machine without a single computational resource being provided. Thus the new architectures will have relatively limited I/O and communication capacity compared with their computational capability.

Enabling computational science on these platforms will require advanced data management capabilities optimized for reducing power costs. Because moving and storing data, even within the memory hierarchy on a single machine, requires significantly more power than computing the data does, reducing these operations will likely become the dominant optimization strategy for HPC codes. Unfortunately, this already difficult task will be further complicated by the trends toward coupled codes, ensemble runs, and in situ analysis identified in other reports. The additional complexity imposed by these concurrently scheduled tasks will make a challenging task even more demanding.

One possible approach to manage this complexity is to utilize in-memory workflows to coordinate the tasks, orchestrate the data movement, and record provenance. This in turn will require new levels of system

support, such as improved schedulers that can co-schedule different tasks and support for data staging. Unfortunately, even successfully managing all of this information is unlikely to ensure bitwise reproducibility of the results, given the expected number of faults occurring in an exascale system and the inability of the simulations to permanently store most of their results. Indeed, as statistical analysis, user-driven analysis, and approximate queries become more common, it is unlikely that any two runs of the same large-scale simulation permanently store exactly the same results, which ultimately represent only a small fraction of the volumes of information generated and analyzed to produce them. This development will drive the need for stronger statistics and uncertainty quantification techniques to determine when simulations are generating consistent results. Developing these specialized capabilities is probably outside the scope of the broad research community, but it is well aligned with ASCR's goals.

Current state-of-the art capabilities work well when applied to the problems they were designed to overcome. Unfortunately, many of the assumptions underlying these technologies will not remain valid at the exascale:

- Files are a traditional mechanism for transferring information between tasks. This will need to change as the cost of writing information to disk then reading it back in will dominate computational costs.
- Workflow engines coordinate tasks within the HPC environment through the scheduler and remote connections to the HPC machine. They do little to help orchestrate tasks that must run concurrently and share memory-based data objects.
- Schedulers are focused on allocation of independent tasks and do not natively support dynamic scheduling based on data flows.
- Memory-based data objects, such as those provided by ADIOS, provide some data staging and movement capabilities, but additional work needs to be done to minimize data transfers between nodes and ensure data remains accessible as long as necessary to complete the tasks.

In addition to breaking fundamental assumptions made by existing technologies, data management at the exascale will incorporate two new sources of complexity: a lack of data coherence and the new NVRAM environment. The cost of maintaining data coherence across all cores in an exascale machine is simply too high. As a result, an exascale computer will not have the memory consistency we currently expect—data changes in one location (core/node) will not necessarily be reflected in other processing elements. How this lack of coherence will be managed by the software algorithms, in which conflicting updates may need to be resolved, is still unclear. What is clear, however, is that developing approaches to accommodate these discrepancies will add a significant data management challenge.

Exascale computers are expected to have an NVRAM layer inserted into the memory hierarchy between DRAM and the global file system. This new memory layer will provide a way to share information between nodes and tasks as well as maintain data persistence for the short term. However, effectively using it may require an application to be explicitly aware of this layer in the hierarchy—an awareness that currently only exists for files—and to manage it, possibly in competition with other tasks and the operating system. Having to explicitly manage this limited resource, including releasing unneeded space and mapping from transient memory to NVRAM, will add a significant level of complexity to applications and I/O middleware.

The conventional approach to addressing this problem is to develop new I/O middleware to enable applications to manage additional complexity. While the current file-based interface provides a consistent, straightforward approach to accessing data, it will not be sufficient for managing multiple levels of the memory hierarchy. It is possible that a new interface capable of managing data across all layers in the hierarchy—sharing objects between tasks, independent of whether the object is in transient memory, persistent memory, or disk—will evolve. An alternative approach, though more challenging and controversial, is to adapt the applications to allow the workflow layer to make control and data movement decisions based on a set of rules and configuration information. This would likely improve the portability and performance of the application code.

At the same time, applications sharing memory-based data objects will increasingly require the ability to specify more complex actions over that data. For example, an interface that provides a higher level of abstraction (e.g., mesh, cell, vertex, variable, time) may allow analysis routines to more effectively identify relevant subsets of the data. Simple analysis steps, such as sampling or aggregating data at regular intervals,

may also become part of this higher-level interface. Finally, schedulers will need to dynamically co-locate tasks that share data in order to minimize data transfers. This will require a new mechanism for specifying the complex data flow dependencies among tasks and their associated cores.

Finally, data provenance has an important role to play in supporting the validation of science at the exascale. While detailed recordings of every event of possible interest would dominate the network and render simulations infeasible, intelligent recording of key parameters and data sets will be crucial to understanding exactly what occurred during the in situ visualization and analysis. This set of information should be small enough to be persistently recorded, yet useful enough that replaying the events will lead to a statistically similar result. Without this level of provenance, the core scientific principle of reproducibility would be lost.

There are a number of approaches under development that may help address projected shortcomings in the exascale data management environment. Unfortunately, these activities tend to be centered within a single research area and do not provide a holistic solution. For example, work on ArrayDB is addressing the need for ad hoc queries of scientific data, and HBase provides a distributed query platform. However, neither of these efforts is being informed by research occurring within the workflow, programming models, or scheduling research communities. Although these individual technologies may ultimately be extremely useful, there is currently no collaborative effort to define the requirements for a comprehensive data management approach, much less active work creating interface standards that projects can use.

Development of these standards, and associated pilot implementations, is an area where ASCR's leadership could have a significant impact on focusing the research community. In particular, defining an environment capable of orchestrating these data management tasks on a leadership-class computer is an excellent opportunity for co-design, since the environment cuts across traditional research areas, from workflow technology to schedulers, from programming models to operating systems, from networking to the memory hierarchy. Because representatives from all of these research areas would be required to effectively address the data management complexity facing exascale scientific application and analysis codes, ASCR is in a unique position to enable development of a meaningful solution.

Ultimately, the transition to exascale computing will transform data management and highlight it as an integral part of large-scale computational science. Data management will move from a file-based focus, where the key metrics reflect disk-based read and write performance, to an integrated framework that supports information sharing and knowledge discovery at all levels of the memory hierarchy. To reach this goal, the traditional data management community will need to work with a much broader research community to address the cross-cutting issues previously identified, develop new data models and abstractions, and effectively manage the complex workflows that will be required to use this new class of machines.

7 Co-design and collaboration opportunities

Although the exascale DMAV workshop was organized under the auspices of DOE ASCR within the Office of Science, we recognize that achieving successful data understanding at the exascale will take a concerted collaborative effort between national laboratories, universities, industry, and other countries, as well as cross-fertilization across federal funding programs. This is no different from the collaborations that have successfully allowed the utilization of HPC resources at the terascale and petascale. However, as the exascale holds unique challenges, there are also unique opportunities for collaboration.

7.1 Hardware vendor collaboration

As hardware vendors impact the design of exascale machines, in turn they impact DMAV; analysis tools will need to run directly on these upcoming systems for certain types of analysis (including in situ processing). In other cases, impacts will be seen indirectly, through the data generated by scientific simulation codes.

DOE, as one of the major deployers of HPC systems, has often partnered with hardware vendors. In particular, scientific applications teams are collaborating with hardware vendors through recent co-design efforts targeting the exascale. Although these applications teams may have in mind only a few limited goals in the analysis and visualization arena, DMAV issues are generally cross-cutting. This raises the question whether the current collaborations with hardware vendors are sufficient to accomplish the needs of DMAV at the exascale.

There are some areas in which partnering with hardware vendors would prove fruitful. As heterogeneous systems rise to the forefront of current HPC systems, partnering with vendors of GPUs and other data parallel devices, as well as with the CPU vendors themselves—both traditional HPC and emerging low-power manufacturers—would lead to greater ability to take advantage of upcoming extreme on-node concurrency. Integrators, system vendors, and interconnect designers could have roles in alleviating off-node concurrency issues, greatly impacting DMAV use cases such as loosely-coupled in situ analysis. As technologies such as NVRAM become integrated in future architectures, discussions with these vendors can help illuminate how to use these new capabilities effectively, not just for scientific application codes but also for the analysis and visualization of their data. Tightly coupled in situ techniques are likely to receive the most significant benefit.

7.2 Software design collaborations

DOE makes heavy use of commodity hardware in the HPC systems it deploys. Commodity software is also heavily used, but mostly at the lower system layers. Much as DOE invests in hardware, it often invests in research and collaboration at the system software level (see the ASCR X-Stack call as a recent example). At the higher levels, the software utilized by DOE is generally more focused on DOE mission goals and is rarely widely available as commodity or commercial applications. This holds true for DMAV software as well. For this reason, partnering with not just industry DMAV software vendors but also with science applications teams, DOE and ASCR institutes, and international efforts such as DEISA (Distributed European Infrastructure for Supercomputing Applications) and PRACE (Partnership for Advanced Computing in Europe), will be crucial for developing effective visualization and analysis capabilities on upcoming HPC systems.

Many visualization and analysis problems are domain-specific and require in-depth knowledge of the domain for an effective solution. The tools to address them often are either designed by visualization programmers with little or no knowledge of the domain, or quickly coded up by an application scientist with limited knowledge of the visualization tools. Tools developed in this fashion tend to be difficult to deploy and maintain.

In response, we require increased collaboration and partnerships with various computer science communities. These include applied mathematics and scalable solver developers who can better understand the computation and physics, as well as application scientists who can better understand their visualization and analysis needs and co-design appropriate solutions. All of these collaborations should extend across government agencies, including the National Science Foundation, the Department of Defense, and the National Institutes of Health, as well as appropriate entities in industry.

8 Conclusion: Findings and Recommendations

The architectural and infrastructure changes coming as we march toward exascale computing are likely to significantly disrupt the current methods for scientific discovery, visualization, analysis, and data movement. Only through a concerted and focused effort toward the adaptation of existing methods and the development of new methods designed for the dramatic limitations of the exascale platforms can we continue to employ large HPC resources for scientific advancement. With this goal in mind, and in light of the research roadmap detailed in the previous sections, we make the following findings and recommendations:

Finding 1: The path to implementing codes targeting architectures comprising hundreds of cores per chip and running at billion-way concurrency is not clear. This challenge is faced by science applications as well as by the DMAV community.

Recommendation 1: Research efforts in DMAV should closely track the evolution of emerging programming environments to evaluate alternative approaches for creating robust and scalable software infrastructure. Also, since the programming language(s)/model(s) of choice at the exascale will be unclear for some time to come, research and development in visualization and analysis could pursue multiple approaches that aim to enable effective use of exascale architectures in both post-processing and in situ/concurrent approaches using these emerging programming models and execution environments.

Finding 2: Because of the growing cost of I/O, it will become increasingly impractical for simulations to perform writes to storage at full spatiotemporal resolution. The traditional post-processing approach by which simulations write data for subsequent analysis and visualization will likely become less common as a result of increasingly expensive I/O. Therefore, it is likely that an increasing amount of analysis and visualization must take place while simulation data is still resident in memory.

Recommendation 2: While there are a few examples of successful in situ visualization going back through the past two decades, future research in this space is needed in several important aspects of the technique to enable its more widespread use in the future. These issues include graceful sharing of resources with simulation code (e.g., memory footprint, cores in a multi-core/many-core environment), minimizing or eliminating data copies in memory, and commoditization of in situ and concurrent visualization and analysis APIs to minimize "one-off" solutions.

Finding 3: Any tightly-coupled in situ solution will share resources with the simulation code. In some situations, this can prohibitively impact the running simulation. Similarly, in situ on the same node exposes the simulation code to an additional source of failure. Though there has been fruitful research and (rarely) production into tightly-coupled in situ analysis, it is not a complete solution.

Recommendation 3: A research effort into in situ data staging is warranted, complementing research into in situ frameworks in general. This allows more resiliency, less resource contention, and opportunities for hardware differentiation. Though communication is greater for in situ data staging than for tightly-coupled in situ, it provides opportunities for balancing priorities.

Finding 4: In situ analysis and visualization is a necessity, but no panacea. Put simply, there is no way to extract knowledge from the deluge of data in an exascale machine without performing some analysis at the origination of the data. However, in situ analysis and visualization exhibits many restrictions that limit exploration and serendipitous discovery.

Recommendation 4: In addition to solving the basic technical problems of coupling simulation with analysis and visualization, in situ technologies must evolve to provide broader analysis and visualization techniques. This evolution will likely involve a blending of in situ processing with post processing. Section 6.1.1 details all these in situ challenges.

Finding 5: Although the relative cost of moving data will remain constant, experts in computer architecture believe that the large amounts of data generated as we head to the exascale regime will make the I/O habits of today prohibitive, especially to and from external storage. Codes from all classes of applications, including scientific simulations as well as visualization and analysis applications, will be affected by this finding.

Recommendation 5: Future research in visualization and analysis should focus on methods that minimize data movement throughout the memory hierarchy, but especially to and from external storage. Similarly, future data management research should focus on techniques to improve the ability of codes to make effective use of the memory and storage hierarchy. Minimizing data movement is a cross-cutting theme throughout this report.

Finding 6: Scientific data formats and file formats are relatively immature within DOE ASCR applications. As the aggregate data set size increases, multi-physics complexity increases, and cross-dataset linking becomes more important, our existing methods for secondary storage will become overly constraining. Similarly, the storage container of a "file" is already inadequate for data sets that cross the petabyte boundary.

Recommendation 6: More research into self-describing formats that provide flexibility, support for high-dimensional fields, and hierarchical mesh types is a pressing need. These formats must be geared for DMAV activities (reading) as well as for writing from simulation codes. See Sections 6.2.6 and 6.3.3 for more details.

Finding 7: Managing raw data from simulations is becoming impractical for I/O networking and processing costs. As the datasets are growing in raw size, and the ratio of I/O bandwidth to FLOP count is decreasing, the ability to write out the raw data from simulations is rapidly being crippled.

Recommendation 7: Data analysis methods should be used as means for massive data reduction that allows saving abstract representations of features in the data instead of the raw bytes generated by the simulations. Research is needed in the parallelization of the analysis computations constrained to low memory usage and to maintaining the data partitioning provided by the simulations so that in situ computation is facilitated. Research is also needed in the development of representations (such as topological or statistical) that achieve the massive data reductions needed while preserving the semantic content needed by scientists to explore and validate hypotheses. Section 6.2 describes the challenges creating data abstractions.

Finding 8: Future architectures will continue to reduce the amount of memory per available compute cycle. Data economy will become synonymous with power economy. Methods of mathematical statistics for optimal use of data will become relevant in a new setting.

Recommendation 8: Developing new techniques in mathematical statistics for optimal use of distributed data should be a part of an interdisciplinary team strategy to address scalable visualization and analysis for future computer architectures. Section 6.2.4 details these challeges.

Finding 9: Although solving exascale-class computational problems is a major and important challenge facing the sciences, much day-to-day science, particularly experimental science, can benefit from advances in DMAV software infrastructure to better support high-throughput experimental needs on current and emerging multi-core/many-core platforms.

Recommendation 9: Research in DMAV can benefit both exascale-class and more modest, experimental-class science. The advances needed to enable DMAV at the exascale will similarly benefit high-throughput, experimental science. Therefore, research and partnerships between DMAV and the sciences should include a diverse cross section of the science community to achieve the broadest possible impact.

Finding 10: Not only is the management of scientific data from simulations necessary, but also the other factors that contribute to generating it (e.g., operating system and compiler version) need to be captured. As datasets continue to increase in size and complexity, assumptions and educated guesses will be made to reduce the dataset size and manage the output. These assumptions and educated guesses need to be captured to provide context and reproducibility. Simulation science needs to move toward more rigor in the capture of metadata and the state of data as it moves toward saving less of it.

Recommendation 10: There is a need to develop *provenance* infrastructure that makes the capture and curation of datasets automatic. It needs to be integrated with both the simulation and the analysis components of the simulation pipeline. On the simulation side, it needs to capture metadata information about the environment and settings of the simulation; and on the analysis side, the captured information needs to be translated into feedback to the user (e.g., the errors introduced by sampling).

Finding 11: A review of the reports from the "Scientific Grand Challenges Workshop Series" reveals a trend toward more complicated, and more complete, physical models using techniques that involve multiple scales, multiple physics, and time-varying data. Also frequently cited are advanced techniques to establish applicability, like uncertainty quantification. Each advance adds a new dimension that is not appropriately handled by current analysis and visualization techniques.

Recommendation 11: Basic research for analyzing and visualizing these new properties is required. Often the techniques for analyzing multiple scales, physics, and comparable runs will be inexorably tied to the particular problem domain. Design of such visualization and analysis requires close collaboration between science domain experts and data analysis experts.

Finding 12: The traditional design of computing is linear and compartmentalized. Hardware is built independently of the software that will run on it. Simulations store results arbitrarily and leave analysis options open-ended. This design strategy is convenient and logical: Dependencies are explicit, management is straightforward, and domain experts can focus on their area of expertise. However, such a design relies on leniency in our basic capabilities. A stable hardware design for general computing capabilities ensures that software can be ported. An ample amount of storage ensures enough space to capture a representative fraction of data. All predictions for exascale computing point to these leniencies being removed.

Recommendation 12: The roadmap to exascale must involve a transition from technology-driven science to discovery-driven science. That is, science must be driven not by the computations that can be performed nor by the physics that can be computed nor by the data that can be generated, but rather driven by the discoveries that need to be made. The end result must be the primary consideration from the very start of the design in computational science. Our first focus must be the discoveries we need to make. These discovery goals drive the analysis and visualization to be performed. The analysis and visualization dictate what data is needed and how it must be managed. The data required and its management dictate what simulations are run, how they are run, and in coordination with which other software facilities they are run. All of these requirements drive hardware design and procurement.

Discovery-driven science challenges all of us in the ASCR community. It commands an unprecedented collaboration among disciplines and projects. Our independent knowledge, tools, and applications must come together in a federated unit to address the cross-cutting issues of exascale that affect us all.

A Appendix: Historical Perspective

This appendix provides a brief overview of research in visualization and data analysis within the DOE community over the last decade and through Scientific Discovery through Advanced Computing (SciDAC) initiatives in the past 5 years. Looking back on the work, a number of themes emerge. In the early days, the research issues centered on data parallel infrastructure for handling large-scale data, and the building and sharing of tools across the scientific community. More recently, the research has shifted to tighter integration of analysis and visualization, data management and scalable I/O, and building domain-specific applications.

In the late 1990s, there were no open-source or commercial visualization packages that could effectively visualize large datasets. This was a significant concern to the scientific simulation community because large-scale results were being generated and needed analysis. A "tri-lab" (Los Alamos, Lawrence Livermore, and Sandia National Laboratories) research initiative was launched to modify, extend, and exploit Kitware's Visualization ToolKit (VTK), an open-source, object-oriented visualization library.

One effort attacked the problem of running VTK in parallel. By modifying the infrastructure of VTK to support data streaming (the ability to incrementally process a dataset), data parallelism, and distributed computing, the new parallel toolkit was able to extend the existing full range of visualization, imaging, and rendering algorithms available in VTK. In addition, parallel rendering algorithms were added. Another effort explored parallelism outside VTK, providing data parallelism in a higher-level library, allowing further optimization for data parallel distributed computing and visualization.

In this way, this powerful VTK foundation was employed by two end-user applications, ParaView and VisIt, to provide scientific visualization applications that scaled from the desktop to the cluster, hiding the details of the parallel configuration and the complexities of the visualization from the scientist. Both application groups subscribed to the idea that moving to open-source software would facilitate the building and sharing of tools across the national laboratories, academia, and industry.

In the early 2000s, explosive growth in the power of commodity graphics cards led to research in using hardware acceleration on commodity clusters. This research was a natural extension of previous parallel, distributed-memory approaches. The programmability of graphics processing units (GPUs) opened up entirely new algorithms not only for visualization but also for various types of analysis. Multiple GPUs also provided the high rendering speeds needed to interactively visualize large data on dedicated graphics clusters.

In 2006, the SciDAC initiative funded three major centers: the Visualization and Analytics Center for Enabling Technologies (VACET), the Scientific Data Management Center (SDM), and the Institute for UltraScale Visualization (IUSV). These centers moved beyond the technology required to interactively view enormous simulation data. Instead, they focused on revealing and understanding deeper relationships found through expanded analysis, integrated visualization, and domain-specific applications.

A.1 VACET

VACET combines a range of visualization, mathematics, statistics, computer and computational science, and data management technologies to foster scientific productivity and insight. It has provided new capabilities to science teams that aid in knowledge discovery. Scientists are able to see, for the first time, features and attributes that were formerly hidden from view. VACET accomplishments range from the development of a production-quality, petascale-capable, visual data analysis software infrastructure that is being widely adopted by the science community as a demonstration of how science is enabled at the petascale, to the scientific impacts resulting from the use of that software.

Specific examples of VACET contributions include

- Topological analysis work that allowed scientists new insight into the fundamental processes of combustion
- A new capability for accelerator researchers to see for the first time all particles that meet a minimum level of being "scientifically interesting" in 3 dimensions and in conjunction with multi-modal visual presentation
- Multi-modal (traditional computational fluid dynamics variables, vector-valued magnetic field, multi-frequency radiation transport) visual data exploration of supernova simulation results from a petascale-class machine

- Very-high-resolution climate models run on petascale-class platforms and containing new features previously not visible, since these massive datasets cannot typically be processed using "standard" desktop visualization applications;
- The new capability for science teams to perform visual data analysis and exploration on uncommon computational grids (e.g., mapped-grid adaptive mesh refinement in fusion science and geodesic grids in climate science) to achieve higher levels of efficiency of parallel platforms
- For DOE's Nuclear Energy program, visual data exploration and analysis infrastructure to study complex flow fields
- Support for effective parallel I/O to a SciDAC climate science team through design and implementation of a data model and parallel I/O library for use on the Cray XT4 platforms at the National Energy Research Scientific Computing Center/LBNL and the Oak Ridge Leadership Computing Facility/ORNL
- Direct support to a SciDAC fusion science center to enable parallel I/O of fusion (and accelerator) simulation data where that effort will "spill over" to help numerous other projects that now use or will use those same simulation codes

A.2 SDM

SDM provides an end-to-end approach to data management that encompasses all stages from initial data acquisition to final data analysis. It addresses three major needs: (1) more efficient access to storage systems through parallel file system improvements that enable writing and reading large volumes of data without slowing a simulation, analysis, or visualization engine; (2) technologies to facilitate better understanding of data, in particular the ability to effectively perform complex data analysis and searches over large data sets, including specialized feature discovery, parallel statistical analysis, and efficient indexing; (3) robust workflow tools to automate the process of data generation, collection and storage of results, data post-processing, and result analysis.

Specific examples of SDM contributions include

- Integration of Parallel NetCDF, successfully used by the large-scale National Center for Atmospheric Research Community Atmosphere Model
- Enhancement of I/O efficiency for the Lustre file system by as much as 400% using partitioned collective I/O (ParColl) without requiring a change in file format
- Development of the Adaptable I/O System (ADIOS), a simple programming interface that abstracts transport information and speeds up I/O on Cray XT, InfiniBand clusters, and IBM Blue Gene/P through the use of a new file format, BP (binary-packed), that is highly optimized for checkpoint operations
- Development of FastBit, an efficient indexing technology (performing 50–100 times faster than any known indexing method) for accelerating database queries on massive datasets. FastBit received an R&D 100 award in 2008.
- Use of FastBit to achieve 1,000 times speedup of particle search for the Laser Wakefield Particle Accelerator project and 1,000 times speedup for identification of gyrokinetic fusion regions
- Development of an open-source library of algorithms for fast, incremental, and scalable all-pairs similarity searches that achieves orders of magnitude (100,000 fold) speedup
- Bringing into production open-source ProRata statistical software, which has been downloaded more than 1,000 times and has been used by the DOE bioenergy centers and Genomics:GTL projects
- Development of an integrated framework, currently being used in production runs by scientists at the Center for Plasma Fusion Edge Simulation, which includes the Kepler workflow system, a dashboard, provenance tracking and recording, parallel analysis capabilities, and SRM-based data movement.

A.3 IUSV

IUSV has introduced new approaches to large-scale data analysis and visualization, enabling scientists to see the full extent of their data at unprecedented clarity, uncover previously hidden features of interest in their data, more easily validate their simulations, possibly interact with their data to explore and discover, and better communicate their work and findings to others. The Institute's research effort is targeted at

the design and evaluation of parallel visualization technology, in situ data triage and visualization methods, time-varying multivariate data visualization techniques, distance and collaborative visualization strategies, and novel visualization interfaces. IUSV also combines a complementary research component and outreach and education component for communication and collaboration. Through working with other SciDAC Institutes, Centers for Enabling Technologies, and application projects, IUSV has already made many research innovations and demonstrations of new technologies. Likewise, through outreach activities, IUSV provides leadership in research community efforts focusing on extreme-scale visualization. These activities include hosting specialized workshops and panels at leading conferences to stimulate widespread participation.

Selected IUSV accomplishments include

- in situ data triage and visualization solutions demonstrated at large scale, driving further development of this technology in the research and SciDAC community
- Parallel volume rendering algorithms scalable to hundreds of thousands of processors, enabling high utilization of the most powerful supercomputer for demanding visualization tasks
- Multi-GPU rendering and visualization libraries for building visualization clusters at different scales
- A framework for data reduction, quality assessment, and LOD (level of detail) visualization facilitating interactive exploration of large data streams
- A set of interactive visualization techniques supported by scalable data servers for browsing and simultaneous visualization of time-varying multivariate data
- Advanced visualization facilities for climate data analysis, including web-enabled collaborative visualization support in the Earth System Grid, a query language for visualizing probabilistic features, and 4-dimensional correlation analysis and visualization
- Advanced study of optimized use of leading-edge parallel I/O in large-scale parallel visualizations
- Transfer of new technologies to end users through ParaView
- A well-attended Ultrascale Visualization Workshop at the annual Supercomputing Conferences over the past 5 years, which highlights the latest large data visualization technologies, fosters greater exchange among visualization researchers and the users of visualization, and facilitates new collaborations
- More than a dozen tutorials in large data analysis and visualization

B Appendix: Workshop Participants

Sean Ahern, Oak Ridge National Laboratory
Jim Ahrens, Los Alamos National Laboratory
Ilkay Altintas, San Diego Supercomputing Center
E. Wes Bethel, Lawrence Berkeley National Laboratory
Eric Brugger, Lawrence Livermore National Laboratory
Surendra Byna, Lawrence Berkeley National Laboratory
C-S Chang, New York University
Jackie Chen, Sandia National Laboratories
Hank Childs, Lawrence Berkeley National Laboratory
Alok Choudhary, Northwestern University
John Clyne, National Center for Atmospheric Research
Terence Critchlow, Pacific Northwest National Laboratory
Ryan Elmore, National Renewable Energy Laboratory
Jinghua Ge, Louisiana State University
Mark Green, Tech-X Corporation
Kenny Gruchalla, National Renewable Energy Laboratory
Chuck Hansen, University of Utah
Jian Huang, University of Tennessee
Keith Jackson, Lawrence Berkeley National Laboratory
Chris Johnson, University of Utah
Ken Joy, University of California–Davis
Chandrika Kamath, Lawrence Livermore National Laboratory

Scott Klasky, Oak Ridge National Laboratory
Kerstin Kleese-van Dam, Pacific Northwest National Laboratory
Quincey Koziol, The HDF Group
Rob Latham, Argonne National Laboratory
Terry Ligocki, Lawrence Berkeley National Laboratory
Gerald Lofstead, Sandia National Laboratories
Kwan-Liu Ma, University of California–Davis
Jeremy Meredith, Oak Ridge National Laboratory
Bronson Messer, Oak Ridge National Laboratory
Steve Miller, Oak Ridge National Laboratory
Kennneth Moreland, Sandia National Laboratories
Lucy Nowell, Department of Energy
George Ostrouchov, Oak Ridge National Laboratory
Mike Papka, Argonne National Laboratory
Manish Parashar, Rutgers University
Valerio Pascucci, University of Utah
Hanspeter Pfister, Harvard University
Norbert Podhorszki, Oak Ridge National Laboratory
Stephen Poole, Oak Ridge National Laboratory
Dave Pugmire, Oak Ridge National Laboratory
Doron Rotem, Lawrence Berkeley National Laboratory
Nagiza Samatova, North Carolina State University
Karsten Schwan, Georgia Institute of Technology
Arie Shoshani, Lawrence Berkeley National Laboratory
Gary Strand, National Center for Atmospheric Research
Xavier Tricoche, Purdue University
Amitabh Varshney, University of Maryland
Jeff Vetter, Oak Ridge National Laboratory
Venkatram Vishwanath, Argonne National Laboratory
Joel Welling, Pittsburgh Supercomputing Center
Dean Williams, Lawrence Livermore National Laboratory
Matthew Wolf, Georgia Institute of Technology
Pak Wong, Pacific Northwest National Laboratory
Justin Wozniah, Argonne National Laboratory
Nick Wright, Lawrence Berkeley National Laboratory
Kesheng Wu, Lawrence Berkeley National Laboratory
Yong Xiao, University of California Irvine

References

[1] James P. Ahrens, Jonathan Woodring, David E. DeMarle, John Patchett, and Mathew Maltrud. Interactive remote large-scale data visualization via prioritized multi-resolution streaming. In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, November 2009.

[2] M. Andrecut. Parallel GPU implementation of iterative PCA algorithms. *Journal of Computational Biology*, 16(11):1593–1599, 2009.

[3] Amit C. Awekar, Nagiza F. Samatova, and Paul Breimyer. Incremental all pairs similarity search for varying similarity thresholds with reduced i/o overhead. In *IKE'09*, pages 687–693, 2009.

[4] Carrie Ballinger and Ron Fryer. Born to be parallel: Why parallel origins give teradata an enduring performance edge. *IEEE Data Engineering Bulletin*, 20(2):3–12, 1997. An updated version of this paper is available at `http://www.teradata.com/library/pdf/eb3053.pdf`.

[5] Dennis A. Benson, Mark S. Boguski, David J. Lipman, James Ostell, B. F. Francis Ouellette, Barbara A. Rapp, and David L. Wheeler. GenBank. *Nucleic Acids Research*, 27(1):12–17, 1999. Data available at `http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html`.

[6] J. Biddiscombe, J. Soumagne, G. Oger, D. Guibert, and J.-G. Piccinali. Parallel computational steering and analysis for HPC applications using a ParaView interface and the HDF5 DSM virtual file driver. In *Proceedings of Eurographics Parallel Graphics and Visualization Symposium*, April 2011.

[7] Roger Blandford, Young-Kee Kim, Norman Christ, et al. Challenges for the Understanding the Quantum Universe and the Role of Computing at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, December 2008.

[8] Peter A. Boncz, Marcin Zukowski, and Niels Nes. MonetDB/X100: Hyper-pipelining query execution. In *CIDR*, pages 225–237, 2005.

[9] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010.

[10] P.-T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. B. Bell. Interactive exploration and analysis of large scale simulations using topology-based data segmentation. *IEEE Trans. on Visualization and Computer Graphics*, page to appear, 2010.

[11] Paul G. Brown. Overview of SciDB: large scale array storage, processing and analysis. In *SIGMOD*, pages 963–968. ACM, 2010.

[12] R. Brun and F. Rademakers. ROOT : An object oriented data analysis framework. *Nuclear instruments & methods in physics research, Section A*, 289(1-2):81–86, 1997.

[13] H. Childs, M. Duchaineau, and K.-L. Ma. A scalable, hybrid scheme for volume rendering massive data sets. In *Proceedings of Eurographics Sympoisum on Parallel Graphics and Visualization*, pages 153–162, 2006.

[14] Hank Childs. Architectural challenges and solutions for petascale postprocessing. *Journal of Physics: Conference Series*, 78(012012), 2007. DOI=10.1088/1742-6596/78/1/012012.

[15] Hank Childs, David Pugmire, Sean Ahern, Brad Whitlock, Mark Howison, Prabhat, Gunther Weber, and E. Wes Bethel. Extreme Scaling of Production Visualization Software on Diverse Architectures. *IEEE Computer Graphics and Applications*, 30(3):22–31, May/June 2010. LBNL-3403E.

[16] Hank Childs, David Pugmire, Sean Ahern, Brad Whitlock, Mark Howison, Prabhat, Gunther H. Weber, and E. Wes Bethel. Extreme scaling of production visualization software on diverse architectures. *IEEE Computer Graphics and Applications*, 30(3):22–31, May/June 2010.

[17] G. S. Davidson, K. W. Boyack, R. A. Zacharski, S. C. Helmreich, and J. R. Cowie. Data-centric computing with the netezza architecture. Technical Report SAND2006-3640, Sandia National Laboratories, 2006.

[18] M. Day, J. Bell, P.-T. Bremer, V. Pascucci, V. Beckner, and M. Lijewski. Turbulence effects on cellular burning structures in lean premixed hydrogen flames. *Combustion and Flame*, 156:1035–1045, 2009.

[19] Jeffrey Dean. Experiences with MapReduce, an abstraction for large-scale computation. In *PACT '06*, pages 1–1, New York, NY, USA, 2006. ACM.

[20] J.J. Dongarra, H. W. Meuer, H. D. Simon, and E. Strohmaier. Top500 supercomputer sites. `www.top500.org`, (updated every 6 months).

[21] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[22] R. Forman. Combinatorial vector fields and dynamical systems. *Math. Z.*, 228(4):629–681, 1998.

[23] B. Fryxell, K. Olson, P. Ricker, F. Timmes, M. Zingale, D. Lamb, P. MacNeice, R. Rosner, and H. Tufo. FLASH: An adaptive mesh hydrodynamics code for modelling astrophysical thermonuclear flashes. pages 131–273, 2000.

[24] Guilia Galli, Thom Dunning, et al. Discovery in Basic Energy Sciences: The Role of Computing at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, August 2009.

[25] Jim Gray, David T. Liu, Maria Nieto-Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. Scientific data management in the coming decade. *SIGMOD Rec.*, 34(4):34–41, 2005.

[26] Jim Gray and Alexander S. Szalay. Where the rubber meets the sky: Bridging the gap between databases and science. *IEEE Data Engineering Bulletin*, 27(4):3–11, December 2004.

[27] Attila Gyulassy, Vijay Natarajan, Mark Duchaineau, Valerio Pascucci, Eduardo M. Bringa, Andrew Higginbotham, and Bernd Hamann. Topologically clean distance fields. *IEEE Transactions on Computer Graphics, Proceedings of Visualization 2007*, 13(6), November/December 2007.

[28] Robert Haimes. pV3: A distributed system for large-scale unsteady CFD visualization. In *AIAA paper*, pages 94–0321, 1994.

[29] Richard Hamming. *Numerical Methods for Scientists and Engineers*. 1962.

[30] Mark Howison, E. Wes Bethel, and Hank Childs. MPI-hybrid parallelism for volume rendering on large, multi-core systems. In *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, May 2010. LBNL-3297E.

[31] C.R. Johnson, S. Parker, and D. Weinstein. Large-scale computational science applications using the SCIRun problem solving environment. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing*, 2000.

[32] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Trans. Visualization and Computer Graphics (TVCG) / Proc.of IEEE Visualization*, 12(5):1052–1060, 2006.

[33] Lawrence Livermore National Laboratory. *VisIt User's Manual*, October 2005. Technical Report UCRL-SM-220449.

[34] Dawn Levy. Exascale Supercomputing Advances Climate Research. *SciDAC Review*, Special Issue, 2010.

[35] J. Lofstead, Zheng Fang, S. Klasky, and K. Schwan. Adaptable, metadata rich IO methods for portable high performance IO. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–10, 2009.

[36] J. Lofstead, F. Zheng, S. Klasky, and K. Schwan. Adaptable, metadata rich IO methods for portable high performance IO. In *Proceedings of IPDPS'09*, May 2009.

[37] K.-L. Ma. In-situ visualization at extreme scale: Challenges and opportunities. *IEEE Computer Graphics and Applications*, 29(6):14–19, November-December 2009.

[38] K.-L. Ma, C. Wang, H. Yu, and A. Tikhonova. In situ processing and visualization for ultrascale simulations. *Journal of Physics (also Proceedings of SciDAC 2007)*, 78, June 2007.

[39] A. Mascarenhas, R. W. Grout, P.-T. Bremer, E. R. Hawkes, V. Pascucci, and J.H. Chen. *Topological feature extraction for comparison of terascale combustion simulation data*. Mathematics and Visualization. Springer, 2010. to appear.

[40] N.M. Master, M. Andrews, J. Hick, S. Canon, and N.J. Wright. Performance analysis of commodity and enterprise class flash devices. In *Petascale Data Storage Workshop (PDSW), 2010 5th*, pages 1 –5, nov. 2010.

[41] Paul Messina, David Brown, et al. Scientific Grand Challenges in National Security: the Role of Computing at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, October 2009.

[42] Kenneth Moreland, Nathan Fabian, Pat Marion, and Berk Geveci. Visualization on supercomputing platform level II ASC milestone (3537-1b) results from Sandia. Technical Report Tech Report SAND 2010-6118, Sandia National Laboratories, September 2010.

[43] Patrick O'Neil and Elizabeth O'Neil. *Database: principles, programming, and performance*. Morgan Kaugmann, 2nd edition, 2000.

[44] M. Tamer Ozsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 2nd edition, 1999.

[45] Tom Peterka, Hongfeng Yu, Robert Ross, Kwan-Liu Ma, and Rob Latham. End-to-end study of parallel volume rendering on the IBM blue gene/p. In *Proceedings of the ICPP'09 Conference*, September 2009.

[46] Tom Peterka, Hongfeng Yu, Robert Ross, Kwan-Liu Ma, and Rob Latham. End-to-end study of parallel volume rendering on the IBM blue gene/p. In *Proceedings of ICPP 09*, Vienna, Austria, 2009.

[47] Milo Polte, Jay Lofstead, John Bent, Garth Gibson, Scott A. Klasky, Qing Liu, Manish Parashar, Norbert Podhorszki, Karsten Schwan, Meghan Wingate, and Matthew Wolf. ...and eat it too: high read performance in write-optimized hpc i/o middleware file formats. In *PDSW '09*, pages 21–25, New York, NY, USA, 2009. ACM.

[48] K. Potter, J. Kniss, R. Riesenfeld, and C.R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum (Proceedings of Eurovis 2010)*, 29(3):823–831, 2010.

[49] Kristin Potter, Andrew Wilson, Peer-Timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *IEEE ICDM Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes, and Impacts*, December 2009.

[50] Robert Rosner, Ernie Moniz, et al. Science Based Nuclear Energy Systems Enabled by Advanced Modeling and Simulation at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, May 2009.

[51] K. Schuchardt, B. Palmer, J. Daily, T. Elsethagen, , and A. Koontz. I/o strategies and data services for petascale data sets from a global cloud resolving model. 78, 2007.

[52] B. A. Shoemaker and A. R. Panchenko. Deciphering protein-protein interactions. 3:e43, 2007.

[53] Amy Henderson Squillacote. *The ParaView Guide: A Parallel Visualization Application*. Kitware Inc., 2007. ISBN 1-930934-21-1.

[54] Rick Stevens, Mark Ellisman, et al. Opportunities in Biology at the Extreme Scale of Computing. Technical report, ASCR Scientific Grand Challenges Workshop Series, August 2009.

[55] Rick Stevens, Andrew White, et al. Architectures and technology for extreme scale computing. Technical report, ASCR Scientific Grand Challenges Workshop Series, December 2009.

[56] Michael Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Samuel Madden, Elizabeth J. O'Neil, Patrick E. O'Neil, Alex Rasin, Nga Tran, and Stanley B. Zdonik. C-store: A column-oriented DBMS. In *VLDB*, pages 553–564, 2005.

[57] Alexander S. Szalay, Ani R. Thakar, and Jim Gray. The sqlLoader data-loading pipeline. *Computing in Science and Engineering*, 10(1):38–48, 2008.

[58] Bill Tang, David Keyes, et al. Scientific Grand Challenges in Fusion Energy Sciences and the Role of Computing at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, March 2009.

[59] Rajeev Thakur, William Gropp, and Ewing Lusk. On implementing mpi-io portably and with high performance. In *In Proceedings of the 6th Workshop on I/O in Parallel and Distributed Systems*, pages 23–32. ACM Press, 1999.

[60] The HDF Group. HDF5 user guide. `http://www.hdfgroup.org/HDF5/doc/UG/index.html`, 2011.

[61] A. Tikhonova, C. Correa, and K.-L. Ma. Visualization by proxy: A novel framework for deferred interaction with volume data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1551–1559, 2010.

[62] A. Tikhonova, Hongfeng Yu, C. Correa, and K.-L. Ma. A preview and exploratory technique for large scale scientific simulations. In *Proceedings of the Eurographics Symposium on Parallel Graphics and Visualization*, 2011.

[63] T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, and D. O'Hallaron. From mesh genertion to scientific visualization: An end-to-end approach to parallel supercomputing. In *Proceedings of ACM/IEEE Supercomputing 2006 Conference*, November 2006.

[64] Unidata. The NetCDF users' guide. `http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/`, 2011.

[65] UniProt. The universal protein resource (UniProt) 2009. *Nucleic Acids Research*, 37:169–174, 2009.

[66] C. von Mering, R. Krause, B. Snel, and *et al.* M. Cornell. Comparative assessment of large scale data sets of protein-protein interactions. 417:399–403, 2002.

[67] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, October 2008.

[68] C. Wang, H. Yu, and K.-L. Ma. Application-driven compression for visualizing large-scale time-varying data. *IEEE Computer Graphics and Applications*, 30(1):59–69, January/February 2010.

[69] Warren Washington et al. Challenges in Climate Change Science and the Role of Computing at the Extreme Scale. Technical report, ASCR Scientific Grand Challenges Workshop Series, November 2008.

[70] Tom White. *Hadoop - The Definitive Guide: MapReduce for the Cloud*. O'Reilly, 2009.

[71] B. Whitlock, J. M. Favre, and J. S. Meredith. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of Eurographics Parallel Graphics and Visualization Symposium*, April 2011.

[72] Robert Wrembel and Christian Koncilia, editors. *Data Warehouses and OLAP: Concepts, Architectures and Solutions*. IGI Global, 2006.

[73] K. Wu, S. Ahern, E.W. Bethel, J. Chen, H. Childs, E. Cormier-Michel, C. Geddes, J. Gu, H. Hagen, B. Hamann, W. Koegler, J. Laurent, J. Meredith, P. Messmer, E. Otoo, V. Perevoztchikov, A. Poskanzer, O. Ruebel, A. Shoshani, A. Sim, K. Stockinger, G. Weber, and W.-M. Zhang. Fastbit: Interactively searching massive data. *J. of Physics: Conference Series*, 180(1), 2009.

[74] Glenn Young, David Dean, Martin Savage, et al. Forefront Questions in Nuclear Science and the Role of High Performance Computing. Technical report, ASCR Scientific Grand Challenges Workshop Series, January 2009.

[75] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K.-L. Ma. In-situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, May-June 2010.

[76] H. Yu, C. Wang, and K.-L. Ma. Parallel volume rendering using 2-3 swap image compositing for an arbitrary number of processors. In *Proceedings of ACM/IEEE Supercomputing 2008 Conference*, November 2008.