LAB 14-1043

FWP 14-017566

Institution: **Sandia National Laboratories**

XVis: Visualization for the Extreme-Scale Scientific-Computation Ecosystem

Project PI: **Kenneth Moreland**, Sandia National Laboratories
Christopher Sewell, Los Alamos National Laboratory
Hank Childs, University of Oregon
Kwan-Liu Ma, University of California at Davis
Berk Geveci, Kitware, Inc.
Jeremy Meredith, Oak Ridge National Laboratory

Mid-year report, FY16

1 Project Description

The XVis project brings together the key elements of research to enable scientific discovery at extreme scale. Scientific computing will no longer be purely about how fast computations can be performed. Energy constraints, processor changes, and I/O limitations necessitate significant changes in both the software applications used in scientific computation and the ways in which scientists use them. Components for modeling, simulation, analysis, and visualization must work together in a computational ecosystem, rather than working independently as they have in the past. This project provides the necessary research and infrastructure for scientific discovery in this new computational ecosystem by addressing four interlocking challenges: emerging processor technology, in situ integration, usability, and proxy analysis.

Emerging Processor Technology One of the biggest recent changes in high-performance computing is the increasing use of accelerators. Accelerators contain processing cores that independently are inferior to a core in a typical CPU, but these cores are replicated and grouped such that their aggregate execution provides a very high computation rate at a much lower power. Current and future CPU processors also require much more explicit parallelism. Each successive version of the hardware packs more cores into each processor, and technologies like hyperthreading and vector operations require even more parallel processing to leverage each core's full potential.

XVis brings together collaborators from the predominant DOE projects for visualization on accelerators and combines their respective features in a unified visualization library named VTK-m. VTK-m will allow the DOE visualization community, as well as the larger visualization community, a single point to collaborate, contribute, and leverage massively threaded algorithms. The XVis project is providing the infrastructure, research, and basic algorithms for VTK-m, and we are working with the SDAV SciDAC institute to provide integration and collaboration throughout the Office of Science.

***In Situ* Integration** Fundamental physical limitations prevent storage systems from scaling at the same rate as our computation systems. Although large simulations commonly archive their results before any analysis or visualization is performed, this practice is becoming increasingly impractical. Thus, the scientific community is turning to running visualization *in situ* with simulation. This integration of simulation and visualization removes the bottleneck of the storage system.

Integrating visualization *in situ* with simulation remains technically difficult. XVis leverages existing *in situ* libraries to integrate flyweight techniques and advanced data models to minimize resource overhead. Within our *in situ* visualization tools, XVis integrates existing visualization algorithms and those incorporating emerging processor technology. XVis also studies the latest techniques for new domain challenges and for post hoc interaction that reconstructs exploratory interaction with reduced data.

Usability A significant disadvantage of using a workflow that integrates simulation with visualization is that a great deal of exploratory interaction is lost. Post hoc techniques can recover some interaction but with a limited scope or precision. Little is known about how these limitations affect usability or a scientist's ability to form insight. XVis performs usability studies to determine the consequences of *in situ* visualization and proposes best practices to improve usability.

Unlike a scalability study, which is always quantitative, XVis' usability studies are mostly qualitative. Our goal is not to measure user performance; rather, we want to learn about the limitations and benefits of incorporating *in situ* methods in scientists' workflows. These studies reveal how the simulation, hardware, and users respond to a particular design and setting.

Proxy Analysis The extreme-scale scientific-computation ecosystem is a much more complicated world than the largely homogeneous systems of the past. There is significantly greater variance in the design of the accelerator architecture than is typical of the classic x86 CPU. *In situ* visualization also yields complicated interactions between the simulation and visualization that are difficult to predict. Thus, the behavior observed in one workflow might not be indicative of another.

To better study the behavior of visualization in numerous workflows on numerous systems, XVis builds proxy applications that characterize the behavior before the full system is run. We start with the design of mini-applications for prototypical visualization operations and then combine these with other mini-applications to build application proxies that characterize the behavior of larger systems. The proxy analysis and emerging processor technology work are symbiotic. The mini-applications are derived from the VTK-m implementations, and the VTK-m design is guided by the analysis of the mini-applications.

2 Progress Report

The XVis research plan specified in the proposal is divided into a set of milestones spread over the 3-year period of the project, divided among the projects research areas, and distributed among the participating institutions. Our report is similarly organized by giving progress on each of these milestones. Our report is abbreviated to include only those milestones with relevant work in the time period of this report.

2.1 Emerging Processors

Milestone 1.a, Initial VTK-m Design (Year 1–SNL, Kitware, ORNL, LANL) Provide the research and design for VTK-m functional operation and, in conjunction with SDAV, develop an initial implementation.

Expected Completion: FY15, Q4 Status: Complete

Milestone 1.a was completed in FY15 Q4. However, development of VTK-m continues throughout XVis, and we continue to report on its progress.

Recent work for VTK-m includes the initial implementation of the "filter" interface, which is the high-level API to run algorithms in VTK-m. We have also implemented a basic rendering library to enable in situ visualization by directly integrating VTK-m.

Much of FY16 Q1 and Q2 was focused on creating a stable and user-friendly release of VTK-m. We expect the first release of VTK-m to by in FY16 Q3.

Additionally, we have implemented a demo application for VTK-m that reads a specified VTK file or generates a uniform structured grid set and then uses VTK-m's Marching Cubes filter to compute the isosurface. After that, the demo uses VTK-m's rendering engine to generate image files using OS Mesa. This demonstration code is used as part of our outreach for teaching VTK-m.

We have also implement new features to in VTK-m including parallel versions of cell-to-point and simple stream line integration.

Milestone 1.b Array Characterization (Year 2–SNL, Kitware) Automatically characterize how arrays are used and leverage that information to optimize memory hierarchy usage.

Expected Completion: FY16, Q4 Status: In Progress

During a design review with NVIDIA engineers we discussed the benefits of using texture memory when accessing global arrays. On NVIDIA hardware, global memory reads must be accessed in 32, 64, or 128 byte transactions. When a warp executes an instruction that uses global memory, the fetches are coalesced into the minimum number of transactions possible. If a warp is well coalesced a single 32, 64, or 128 byte transaction will suffice, otherwise more transactions will occur causing throughput to suffer. Texture memory is global memory backed by the L1 texture cache, allowing for higher throughput when there is 2D locality of the memory fetches.

In VTK-m uncoalesced memory access are very common when doing any algorithm that requires two different types of topological information, for example cells, and points or faces and edges. To solve this problem we have implemented custom classes that wrap all memory reads when executing on CUDA. These classes then use the provided CUDA command ldg allowing for texture memory reads from global memory accesses without explicitly constructing texture objects. This has resulted in a ~10% performance increase when executing Cell based algorithm that require Point based global memory reads.

Milestone 1.c Hybrid Parallel (Year 2–LANL) Compare alternative models for the interaction of shared-memory and distributed-memory parallelism within VTK-m.

Expected Completion: FY16, Q4 Status: In Progress

In collaboration with Hamish Carr from the University of Leeds, we explored the interaction between shared-memory data-parallelism and inter-node distributed-memory parallelism in the context of an algorithm for computing contour trees (Reeb graphs). Contour trees encode the topological changes that occur to the contour as the isovalue ranges between its minimum and maximum values. They can be used to identify the most "important" isovalues in a data set according to various metrics (e.g., persistence). Although topological analysis tools such as the contour tree and Morse-Smale complex are now well established, there is still a shortage of efficient parallel algorithms for their computation, in particular for massively data-parallel computation on a SIMD model. We developed a novel data-parallel algorithm for computing the fully augmented contour tree using a quantized computation model. We then extended this to provide a hybrid data-parallel / distributed algorithm, allowing scaling beyond a single GPU or CPU, and tested its scaling using Earth elevation data from GTOPO30 across 16 nodes. Our implementation uses the portable data-parallel primitives provided by Nvidia's Thrust library, as well as MPI for inter-node communication. A paper describing this initial algorithm was released as a tech report (LA-UR-15-24759). More recently, in collaboration with Hamish Carr as well as Gunther Weber from LBNL, we have helped develop a new data-parallel contour tree algorithm that does not quantize the contour values, allowing for more precise results and much less memory usage. Preliminary results indicate speed-up on the order of 5x with OpenMP and 20x on GPUs compared to the serial algorithm. We are currently tentatively planning an LDAV submission on this new algorithm.

We have also initiated a collaboration with the project "A Unified Data-Driven Approach for Programming In Situ Analysis and Visualization", led by Pat McCormick, to evaluate the performance of the prototype integration of VTK-m with Legion that they have developed. Legion is a task-parallel runtime that can schedule tasks using a task graph based on the dependencies between the tasks. This is an alternative to the traditional bulk-synchronous MPI model, as used by VTK. We have successfully compiled the code produced by McCormick's project on the Moonlight supercomputer at Los Alamos, using GASNet, OSMesa, VTK-m, and Legion. We have run their isosurface example across four nodes, and are starting to see some performance scaling. Over the next six months, we plan to perform a scaling study across up to several hundred nodes, and compare this to VTK-m used within the VTK MPI-based pipeline. If time permits, we may also help extend the code so that it can schedule some VTK-m tasks on CPUs and some on GPUs during the same run.

2.2 In Situ

Milestone 2.b Post Hoc Interaction (Year 1–U Oregon) Implement three algorithms that use extreme-scale features such as non-volatile memory or knowledge of communication efficiencies.

Expected Completion: FY15, Q4 Status: Complete

We have made significant progress on this milestone in FY16. Our effort has focused on deep memory hierarchies, specifically the SSDs appearing increasingly often on leading-edge supercomputers. Following the "in situ reduction+post hoc" paradigm in collaboration with Childs' Early Career award, we wanted to explore the opportunities available from having significantly more memory for storing data. In particular, using that memory to store multiple time slices and then compressing the data to take advantage of temporal coherence. Our experiments specifically focused on wavelet compression. While wavelet compression typically operates on one time slice at a time (3D data), our study also included multiple time slices (4D data). Our findings showed that the 4D approach could take advantage of temporal coherency, and, for all metrics studied, the benefits were approximately a factor of two improvement. We submitted a paper detailing the experiments to the IEEE Visualization conference and it is in review. In terms of our milestone, we believe this study provides evidence that handling for deep memory/SSDs should be added to VTK-m. Further, we are now planning on adding wavelet compression and decompression operators to VTK-m for general usage.

Milestone 2.b explores architectural features currently beyond the expressivity of VTK-m. For this milestone, three architectural features are to be explored and evaluated, to potentially influence the VTK-m design. We have made excellent progress on our first architectural feature, which explores the performance improvements possible when using different types of memory. We performed this milestone within our ray-tracing code (now ported to VTK-m) and found that accessing GPU-specific memory significantly improved performance. This finding was a contributor in the expansion of VTK-m's improved memory. For our second architectural feature, we are exploring the usage of SSD for post-hoc exploration. A study is underway, which we expect to complete in the next three months. Finally, we decided to focus on deep memory hierarchies for the third architectural feature, and also to delay this study until more architectures are available (i.e., NVLink). This delay is consistent with our under spending and should not interfere with the completion of the project.

Milestone 2.c Flyweight In Situ (Year 2–Kitware) Provide flyweight in situ visualization techniques into a feature-rich, general-purpose library.

Expected Completion: FY16, Q4 Status: In progress

Kitware has been investigating using non-standard memory layouts for arrays and data structures. As a first step towards this goal, we developed the MappedDataArray and MappedDataSet classes, which allow for custom memory layouts. After further evaluation, our conclusion was the overhead introduced by the abstraction used in this approach is too high. We developed and integrated the next generation version of this framework including the classes AOSDataArrayTemplate and SOADataArrayTemplate into VTK. This design depends on template-based polymorphism, and gives us performance that is close to using raw pointers while attaining the objective of allowing tight coupling of VTK in situ with simulations. The updated design allows for things such as constant value arrays, implicit point arrays, and other efficient data model concepts that VTK-m also has. Currently we are investigation how this updated design can allow for seamless and efficient use of VTK-m within VTK with no memory copies.

As part of our collaboration with NVIDIA we developed a prototype that demonstrated VTK-m integration inside VTK and ParaView Catalyst for in situ analysis at Supercomputing 2015. The prototype used the PyFr simulation running on 256 nodes of Titan. The entire operation from simulation computation, visualization algorithms such as IsoSurface, to rendering happened completely on the GPUs and required no memory copies of PyFr simulation data.

Milestone 2.d Data Model Application (Year 2–ORNL) Explore application of new data models to novel architectures appropriate to in situ.

Expected Completion: FY16, Q3 Status: In progress

We have been exploring the integration of VTK-m in situ with several applications running on DOE LCFs. These have included several fusion codes, and a computational seismology code. Efforts have been focused on understanding the scientific workflows being used by these applications. This better allows us to target machine architectures and analysis products for use in situ to help scientists understand their simulations. Included in these efforts is work being done with XGC, a highly scalable physics code used to study plasmas in fusion tokamak devices. Recent work has explored using light-weight plugins to perform visualization both of the particles and the field variables using ADIOS and DataSpaces.

Milestone 2.e Memory Hierarchy Streaming (Year 2–LANL) Develop streaming out-of-core versions of key visualizations and analysis algorithms to efficiently use deep memory hierarchies within in situ applications.

Expected Completion: FY16, Q4 Status: In Progress

We have begun to experiment with the use of the STXXL library from Karlsruhe University for streaming data from disk into main memory and into accelerator memory for isosurface and KD-tree construction algorithms. We have also prototyped the combination of such external-memory algorithms with our distributed wrapper for Thrust as a first step towards enabling these algorithms to operate on data that is both distributed across nodes and too large on each node to fit into memory. We have plans to have a graduate student intern to work on this milestone this summer. His work may involve evaluating the performance of VTK-m as data streams through the various levels of the memory hierarchy on GPUs and/or multi-core CPUs. As we further explore the use of VTK-m within Legion under Milestone 1.c, there may also be opportunities to explore how data streams through VTK-m tasks operating under Legion.

Milestone 2.f Interface for Post Hoc Interaction (Year 2–U Oregon) Define an abstract VTK-m interface for extreme-scale, implement a prototype for the interface, and exercise the prototype with the three algorithms chosen for Milestone 2.b.
Expected Completion: FY16, Q4 Status: Delayed

We have not yet begun this milestone. We expect to be delivering, but the delay is consistent with the our spending rate, and this milestone is not on the critical path of the project.

2.3 Usability

Milestone 3.b Prepare Usability Studies (Year 2–UC Davis, U Oregon) Identify participants and meet to discuss goals for the visualization and analysis tasks. Collect user data sets and design each user study according to code and hardware settings.
Expected Completion: FY16, Q2 Status: Complete

University of Oregon Ph.D. student James Kress has embedded with the Oak Ridge visualization team (including mentorship from Dave Pugmire) in an effort to engage the XGC team. Over the last six months, he has conducted over twenty interviews to understand the XGC team's data needs. This study has included physicists, computational scientists, and computer scientists. Information obtained ranges from predicted data sizes to desired visualization and analysis operations. The next phase of this study has used this information to consider in situ feasibility. The desired visualization and analysis operations have been grouped into approximately five equivalence classes, and we are doing performance modeling for these classes, grounded by real-world experiments on Oak Ridge's supercomputers. We believe the outcome of this work will inform in situ feasibility for the XGC team, and also have implications for other code teams. This work continues into milestone 3.d.

Milestone 3.c Start Usability Studies (Year 2–UC Davis) Conduct pilot studies with algorithms and participants identified in Milestone 3.b.
Expected Completion: FY16, Q4 Status: In progress

In the area of combustion, we are currently working with Dr. Jackie Chen's research group at Sandia National Labs. We have been developing analysis and visualization tools to solve specific analysis problems posed by Dr. Chen, and we will study the usability of these tools. We are focusing on identifying the most effective way of deploying each tool to seamlessly integrate with the S3D simulation code and be able to conduct analysis with very little modification to scientists' current workflow. For example, a histogram-based particle selection scheme is used in one tool to support particular quantitative analysis needs. A set of spatially distributed histograms (probability distribution functions) are generated in situ using the high resolution field data that is available. In addition, a separate module organizes particle data according to the histogram geometry before it is saved to disk. A post-processing visualization system is then used to explore histograms throughout the spatial domain. Users select sets of histograms based on desired distributions in their values. These histograms are then used to identify and fetch large groups of particles in an efficient manner for use in later post-hoc analysis tasks.

So far, the in situ modules have been completed and integrated into a version of the S3D simulation code. We are currently taking steps to ready the code for testing in a large-scale production run. In addition, the post-processing visualization tool is being developed and will run remotely at the supercomputing facility to minimize data movement needs. Once completed, we will evaluate the usability of our tools and methods used for integration and make adjustments as necessary. A submission will be made to LDAV 2016 describing the ongoing progress of this work.

In the area of cosmology, we are working with Dr. Salman Habib's research group at the Argonne National Laboratory to develop and deploy an interactive visualization tool that can effectively use the parallel GPU clusters available to them. We have developed GPU accelerated, parallel rendering methods for interactive visualization of both the particle data and merger tree data. We have also incorporated a few quantitative analysis functionalities. An early version of this tool has been reported in a paper presented in April at PacificVis 2016. Our current effort is to further enhance the scalability of the system. The first deployment is expected to be done in June, and subsequently usability studies will be carried out with full participation of the scientists at Argonne.

2.4 Proxy Analysis

Milestone 4.a Initial Mini-App Implementation (Year 1–SNL, ORNL) An initial implementation of mini-applications based on visualization and in situ workloads.
Expected Completion: FY15, Q4 Status: In progress, delayed

Although milestone 4.a was scheduled to be started at the beginning of the project, the majority of the work has been postponed in lieu of providing a VTK-m prototype (milestone 1.a), which is on the critical path.

The initial prototype for the Marching Cubes mini-app was implemented in FY15, Q4. The implementation will be hardened and contributed to Mantevo in FY16, Q3. This will give us an implementation to start working on subsequent milestones. A mini-driver for rendering and a mini-app for particle advection will follow soon after. This delay is consistent with our under spending and should not interfere with the completion of the project.

Milestone 4.b Validate Mini-App Characteristics (Year 2-ORNL) Validate behavior and resource usage of mini-applications against that of real applications and generate performance/resource models.
Expected Completion: FY16, Q4 Status: In progress

In preparation for proxy analysis, we have been familiarizing ourselves with the Oxbow suite of application characterization tools and have performed some initial within-node characterization of contouring algorithms in existing tools. For example, an early comparison of a sequential contouring algorithm in VisIt and a data-parallel contouring algorithm in EAVL, one of the predecessor projects to VTK-m, we noticed that while there is no thread-level parallelism in VisIt, it was able to make use of integer SIMD arithmetic, while the highly-parallel EAVL algorithm was not.

In this period, we have begun exploring the characteristics of the Marching Cubes mini-app (developed as part of Milestone 4.a) with the Oxbow suite. These early investigations show similarities and differences with the EAVL version of the algorithm. For example, in terms of instruction mix both have a high proportion of integer operations (about 40%) and memory operations (about 25%), but algorithmic choices resulted in different tradeoffs between branch operations and memory movement operations. In memory bandwidth behavior, both had similar read bandwidths, but a noticeable difference in write bandwidths. This preliminary data is shown in the following table, and continuing studies will explore further characteristics and compare the proxy app behavior with the recently developed VTK-m algorithm.

	Instruction Mix (%)				Memory Bandwidth			
Application	BrOps	IntOps	MemOps	Moves	Read B/Cycle	Read MB/s	Write B/Cycle	Write MB/s
Mini-app	20.35	40.06	27.60	11.01	0.07	240.76	0.02	63.18
EAVL	8.76	38.99	24.62	27.12	0.10	283.92	0.04	109.62

3 Other Activities

3.1 Publications

"An Integrated Visualization System for Interactive Analysis of Large, Heterogeneous Cosmology Data," Annie Preston, Ramyar Ghods, Jinrong Xie, Franz Sauer, Nick Leaf, Kwan-Liu Ma, Esteban Rangel, Eve Kovacs, Katrin Heitmann, Salman Habib. In *Proceedings of IEEE PacificVis*, April 2016.

"The Tensions of In Situ Visualization." Kenneth Moreland. *IEEE Computer Graphics and Applications*, 36(2), March/April, 2016.

"Visualization Techniques for Studying Large-Scale Flow Fields from Fusion Simulations." Franz Sauer, Yubo Zhang, Weixing Wang, Stéphane Ethier, Kwan-Liu Ma, *Computing in Science and Engineering*, 18(2), 68–77, March/April 2016.

"Visualization for Exascale: Portable Performance is Critical." Kenneth Moreland, Matthew Larsen, and Hank Childs. *Supercomputing Frontiers and Innovations*, 2(3), 2015.

"ParaView Catalyst: Enabling In Situ Data Analysis and Visualization." Utkarsh Ayachit, Andrew Bauer, Berk Geveci, Patrick O'Leary, Kenneth Moreland, Nathan Fabian, Jeffrey Mauldin. In *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV 2015)*, November 2015.

"Integrated explorer for cosmological evolution." Annie Preston, Franz Sauer, Ramyar Ghods, Nick Leaf, Jinrong Xie, Kwan-Liu Ma, In *IEEE Scientific Visualization (SciVis)*, 107–114, October 2015.

"Volume rendering with data parallel visualization frameworks for emerging high performance computing architectures." Hendrik A. Schroots, Kwan-Liu Ma, In *SIGGRAPH Asia Visualization in High Performance Computing*, 3:1–3:4, November 2015.

"High performance heterogeneous computing for collaborative visual analysis." Jianping Li, Jia-Kai Chou, Kwan-Liu Ma, In *SIGGRAPH Asia Visualization in High Performance Computing*, 12:1–12:4, November 2015.

"Scalable Parallel Distance Field Construction for Large-Scale Applications." Hongfeng Yu, Jinrong Xie, Kwan-Liu Ma, Hemanth Kolla, Jacqueline H. Chen, *IEEE Transactions on Visualization and Computer Graphics*, 21(10), 1187–1200, October 2015.

"In situ depth maps based feature extraction and tracking." Yucong Chris Ye, Yang Wang, Robert Miller, Kwan-Liu Ma, Kenji Ono, In *Large Scale Data Analysis and Visualization (LDAV)*, 1–8, October 2015.

"Fast uncertainty-driven large-scale volume feature extraction on desktop PCs." Jinrong Xie, Franz Sauer, Kwan-Liu Ma, In *Large Scale Data Analysis and Visualization (LDAV)*, 17–24, October 2015.

"Scalable visualization of discrete velocity decompositions using spatially organized histograms." Tyson Neuroth, Franz Sauer, Weixing Wang, Stéphane Ethier, Kwan-Liu Ma, In *Large Scale Data Analysis and Visualization (LDAV)*, 65–72, October 2015.

3.2 Chairs

Symposium Co-Chair, *Large Scale Data Analysis and Visualization (LDAV)*, Hank Childs, October 23, 2016.

Papers Co-Chair, *IEEE Information Visualization (InfoVis)*, Kwan-Liu Ma, October 23–28, 2016.

Papers Co-Chair, *Large Scale Data Analysis and Visualization (LDAV)*, Kenneth Moreland, October 23, 2016.

Papers Co-Chair, *EuroVis*, Kwan-Liu Ma, June 6-10, 2016.

Workshop Co-Chair, *The 10th Workshop on Ultrascale Visualization*, Kwan-Liu Ma, SC15, November 16, 2015.

Papers Co-Chair, *Large Scale Data Analysis and Visualization (LDAV)*, Hank Childs, October 25–26.

3.3 Committees

Program Committee, *12th International Symposium on Visual Computing (ISVC)*, Kenneth Moreland, December 12–14, 2016.

Program Committee, *IEEE Scientific Visualization (SciVis)*, Kenneth Moreland, October 23–28, 2016.

Program Committee, *IEEE Cluster*, Kenneth Moreland, September 12–16, 2016.

Program Committee, *EuroVis*, Kenneth Moreland, June 6–10, 2016.

Program Committee, *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, Kenneth Moreland, June 6–7, 2016.

Program Committee, *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, Hank Childs, June 6–7, 2016.

Program Committee, *IEEE Pacific Visualization (PacificVis)*, Hank Childs, April 20–23, 2016.

Program Committee, *SPIE Visualization and Data Analysis*, Hank Childs, February 16–18, 2016.

NSF III 2016 Review Panel, Kenneth Moreland.

Program Committee, *ACM/IEEE Supercomputing*, Hank Childs, November 15–20, 2015.

Program Committee, *Visual Performance Analysis* (SC workshop), Hank Childs, November 20, 2015.

Program Committee, *IEEE Scientific Visualization (SciVis)*, Kenneth Moreland, October 25–30, 2015.

Program Committee, *IEEE Scientific Visualization (SciVis)*, Hank Childs, October 25–30, 2015.

Program Committee, *IEEE Cluster*, Kenneth Moreland, September 8–11, 2015.

Program Committee, *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, Kenneth Moreland, May 25–26, 2015.

3.4 Presentations and Other Outreach

"Visualizing Extreme Scale CFD Simulations," Plenary speech, Kwan-Liu Ma, *Parallel CFD 2016 Conference*, May 11, 2016.

"Visualization: A Tool for Data Exploration and Storytelling," Invited talk, Kwan-Liu Ma, Taipei Medical University, Taiwan, April 28, 2016.

"The In Situ Terminology Project," Hank Childs, *Department of Energy Computer Graphics Forum (DOECGF)*, April 28, 2016.

"Recent Advances in Visualization Research," Invited talk, Kwan-Liu Ma, Institute of Sociology, Academia Sinica, Taiwan, April 27, 2016.

"Big Data Visualization," Invited talk, Kwan-Liu Ma, *Summit Forum on Big Data Visualization*, Fudan University, Shanghai, China, April 14, 2016.

"Adapting the Visualization Toolkit for Many-Core Processors with the VTK-m Library." Presentation, Christopher Sewell and Robert Maynard, *GPU Technology Conference*, April 2016.

"Topics in Visualization," Invited talk, Institute for Visualization and Interactive Systems, Kwan-Liu Ma, University of Stuttgart, Germany, March 11, 2016.

"Exploratory and Explanatory Visualization," Keynote speech, Kwan-Liu Ma, *3rd EMBO Conference on Visualizing Biological Data (VIZBI)*, March 9, 2016.

"Exascale Visualization: What Will Change." Invitied talk, National Center for Atmospheric Research, Boulder, CO, March 2016.

"XVis, VTK-m, and the ECP," Kenneth Moreland, Data/Vis Panel for the Exascale Computing Initiative Project, February 19, 2016.

"Data Visualization," Invited talk, Kwan-Liu Ma, *Medical Health Informatics*, UCDMC, Sacramento, CA, November 25, 2015.

"Visualization and High Performance Computing," Kwan-Liu Ma, Keynote speech, Symposium on Visualization in HPC, SIGGRAPH Asia, November 2, 2015.

"Advanced Concepts and Strategies for Visualizing Large-Scale, Complex Simulation Data," Kwan-Liu Ma, Invited Talk, *International Computational Accelerator Physics Conference (ICAP)*, October 14, 2015.

"Exascale Visualization: Get Ready for a Whole New World," Hank Childs, Invited talk, *International Computing for the Atmospheric Sciences Symposium (iCAS2015)*, Annecy, France, September 2015.

"VTK-m," Jeremy Meredith, FASTMath PI Meeting, September 2015.

"New Techniques for Visualizing Large-Scale Scientific Data," Kwan-Liu Ma, Invited talk, Software Center for High Performance Numerical Simulation, Chinese Academy of Engineering Physics, Beijing, China, September 2, 2015.

VTK-m Code Sprint, LLNL, September 1-2, 2015.

"VTK-m Overview," Kenneth Moreland, VTK-m Code Sprint, September 1, 2015.

"Trends and Advanced Concepts for Scientific Visualization," Kwan-Liu Ma, Keynote speech, China Scientific Data Conference, August 26, 2015.

"VTK-m: Accelerating the Visualization Toolkit for Multi-core and Many-core Architectures," Christopher Sewell, et al., SciDAC PI Meeting (poster), July 2015.

"VTK-m," Kenneth Moreland, DOECGF, April 2015.

"Hands-on Lab: In-Situ Data Analysis and Visualization: ParaView, Catalyst and VTK-m," Marcus Hanwell and Robert Maynard, GTC Lab, March 2015.

"Visualization Toolkit: Faster, Better, Open Scientific Rendering and Compute," Robert Maynard and Marcus Hanwell, GTC Presentation, March 2015.

"Roadmap for Many-Core Visualization Software in DOE," Jeremy Meredith, GTC Presentation, March 2015.

4 Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract number 14-017566.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

U.S. DEPARTMENT OF ENERGY Sandia National Laboratories

SAND 2016-4942 R