SAND2017-11864R

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
David Pugmire, Oak Ridge National Laboratory
David Rogers, Los Alamos National Laboratory
Hank Childs, University of Oregon
Kwan-Liu Ma, University of California at Davis
Berk Geveci, Kitware, Inc.

Year-end report, FY17

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.

The use of VTK-m has increased significantly during FY17. The most significant development is the award of an ECP project to update VTK software using the VTK-m framework. This demonstrates a large success of XVis: the VTK-m software is being adopted as a key component for our exascale visualization software.

During the first quarter of FY17 the VTK-m accelerator module was added to VTK. This module provides both the infrastructure to allow VTK filters to efficiently use VTK-m filters, and an initial set of VTK-m accelerated filters including: Threshold, Contour, Gradient, and Level of Detail. At the same time ParaView was updated to expose these new VTK-m accelerated filters. ParaView 5.3 is the first official release of ParaView that includes VTK-m accelerated filters.

On April 4-6, 2017 we held our third VTK-m code sprint. It was hosted at the University of Oregon and about 20 people participated. Notable activities included launching work for particle tracing, exploring the performance of hash functions, and looking into the vectorization of VTK-m code.

We also have explored support for graph data in VTK-m. This work was motivated by an EOD use case from Lawrence Berkeley. Our results show reasonable performance for a compute-intensive graph algorithm (maximal cliques). Overall, this direction could create a new class of stakeholders for VTK-m. The results of the work were published at LDAV17.

Finally, we have looked at performance anomalies over architectures. This work was motivated by earlier work on external facelist calculate (published at EGPGV16) and observations that some algorithms performed unusually over different architectures. The resulting study led to better understanding of best practices for hashing, and was also published at LDAV17.

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: Complete

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 algorithm, "Hybrid Data-Parallel Contour Tree Computation", was presented at the Computer Graphics and Visual Computing Conference (CGVC) in September 2016.

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 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. However, issues with global dynamic memory allocation used in virtual mapping for Legion have slowed development in McCormick's project, and so their code is not yet ready for us to run a large scaling study. We anticipate that they will be able to resolve these issues within the next month, and that we will be able to begin to obtain some meaningful performance results in the upcoming quarter.

Milestone 1.d Additional Algorithms (Year 3–LANL) Develop algorithms for additional visualization and analysis filters in order to expand the functionality of the VTK-m toolkit to support less critical but commonly used operators.

Expected Completion: FY16, Q4 Status: Complete

In collaboration with Hamish Carr as well as Gunther Weber from LBNL, we have helped develop and implement a new data-parallel contour tree algorithm that does not quantize the contour values, allowing for more precise results and less memory usage. Our shared SMP algorithm for parallel contour tree computation has formal guarantees of O(log(n) log(t)) parallel steps and O(nlog(n)) work. It employs "parallel peak pruning", in which superarcs are created in the join tree by identifying peaks and finding their governing saddles, and recursively pruning the regions for each peak/saddle pair. We implemented this algorithm natively in OpenMP and with the Thrust library, achieving up to 10x parallel speed up on multi-core CPUs and up to 50x speed up on Nvidia GPUs compared to the serial version. A paper about this algorithm, "Parallel Peak Pruning for Scalable SMP Contour Tree Computation", was accepted to the 2016 IEEE Symposium on Large Data Analysis and Visualization, and it received the Best Paper award. Performance was compared to the serial sweep-and-merge algorithm published by Carr in 2003, which was also used to verify the results. Pat Fasel at Los Alamos has converted the Thrust implementation to use VTK-m, and the implementation was merged into the master branch of VTK-m repo in Jan 2017.

Samuel Li, a graduate student working with Chris Sewell at Los Alamos over the summer, has implemented 1D and 2D wavelet transformation worklets in VTK-m. A set of worklets are used to support the wavelet calculation, including boundary handling, decomposition of signals into approximation and detail coefficients, and reconstruction of the original signal from the wavelet coefficients. Coefficients resulting from the wavelet transforms can be used for compression. The simplest strategy is to threshold small coefficients, which has been implemented and achieves fairly good compression. These calculations can be applied recursively to the output to obtain better compression. This implementation of wavelet transforms uses filter banks, which has the advantage of flexibility to support more wavelet filters. Currently it supports four widely used wavelet filters for compression: CDF 9/7, CDF 8/4, CDF 5/3, and HAAR. He has attempted to minimize the number of data transfers and copies required by the algorithm, and is currently comparing the performance and accuracy of the data-parallel VTK-m algorithm to domain decomposition parallelization strategies as used by the wavelet compression algorithms in VAPOR. By applying the transformation to each element of the input with data-parallelism, it is expected that greater accuracy can be obtained than by using domain decomposition for the same amount of computation, with at least some types of input (such as Gaussians). The VTK-m implementation also allows these same worklets to run on multiple architectures, including GPUs. The 1D wavelet compression worklets have already been merged into VTK-m, while we expect to do the same for the 2D worklets within the next few weeks. Samuel finished this work in Oct. 2016 and this has been merged into VTK-M. A paper titled "Achieving Portable Performance For Wavelet Compression Using Data Parallel Primitives" has been submitted to EGPGV 2017.

Milestone 1.e Function Characterization (Year 3–SNL, Kitware) Design methods to characterize how functions behave and leverage this information for heterogeneous architectures.
Expected Completion: FY16, Q4 Status: In Progress

We have investigated how different functional structures effect vectorization on Intel-based architecture. We have found that for a surprising number of function calling parameters, a technique called loop tiling can have dramatic impact on the efficiency of the execution and the effectiveness of vectorization. In loop tiling for loops that iterate an operation over some set of arrays is broken into a nested inner loop of fixed size (say 1024 items). This minor change can have surprising effect on both the compiler and the hardware (caching and vectorization) units. Additionally, in some circumstances we found that while engaging loop tiling we could transform the structure of data (e.g. from AoS to SoA) without incurring a performance penalty.

Our more recent work includes exploring the idea of adding this loop tiling to the scheduling within VTK-m. We verified that loop tiling within the worklet scheduling within VTK-m can indeed improve performance, and the optimal tiling includes an inner loop of 512 to 1024 items. We are currently working on using this loop tiling and block buffering to try to optimize he vectorization that compilers can achieve. We are integrating this work into VTK-m in such a way that none of the software that it affects needs to be changed and are in the process of widening the performance tests.

During FY17 we have researched how to design efficient polymorphic runtime classes that work across numerous accelerator systems. This has been driven by the need to have algorithms whose components can be switched at runtime such as which integrator to use for streamlines or what function to clip by. The introduction of runtime polymorphism also offers VTK-m the ability to reduce binary size without compromising on performance. The result of this research has been the discovery and application of a pattern very similar to C function callbacks, that works on all accelerator devices, and doesn't compromise performance. This design has been used to improve the implicit function support in VTK-m, and as part of a redesign of the scheduling infrastructure in VTK-m, which resulted in significant binary size reduction.

2.2 In Situ

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: Complete

We have demonstrated the integration of VTK-m into the Sensei and Alpine in situ frameworks. Both of these frameworks aim to provide general purpose lightweight in situ capability. Sensei aims to facilitate the instrumentation of simulation codes for in situ processes while supporting a diverse set of frameworks including Catalyst, Libsim and ADIOS. In addition, it provides an interface that can be integrated with VTK-m for a lightweight in situ solution. It has been demonstrated to scale to thousands of MPI ranks.

Milestone 2.e Memory Hierarchy Streaming (Year 2–LANL) Develop streaming out-of-core versions of key visualization and analysis algorithms to efficiently use deep memory hierarchies within in situ applications.
Expected Completion: FY16, Q4 Status: Complete

In follow-on work to last year's milestone, Chris Sewell and Li-Ta Lo at LANL in collaboration with Tom Fogel from Nvidia have implemented Unified Memory support for Nvidia's Pascal GPUs. This enables VTK-m to seamlessly process datasets larger than GPU memory. We explored several different memory cache policies for performance tuning. A preliminary performance benchmark shows a range of 1x–4x performance impact depending on the complexity of the visualization operation.

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: In Progress

We have completed two of the three algorithms. The first, wavelet compression, has been implemented in VTK-m, and undergone extensive performance testing. The code is available in the VTK-m repository, and the paper documenting the research behind this work was published at EGPGV17. The work showed that VTK-m performance is comparable with architecture-specific implementations, in particular the VAPOR code from NCAR. The second algorithm is rendering, which enables CINEMA-style compression. Matthew Larsen completed many rendering algorithms while he worked as part of the XVis project. We had originally considered Lagrangian flow for the third algorithm, but that work is now being supported by ECP. We have been underspending on our budget, and the work completed to date is consistent with our underspend. We have requested a no-cost extension to implement the remaining algorithm. We are targeting subsampling and multi-resolution techniques.

Milestone 2.g Data Model Array Characterization (Year 3–ORNL) Extend the data model to support array characterizations from Milestone 1.b for supporting heterogeneity.
Expected Completion: FY17, Q3 Status: Complete

We have completed this milestone by having the data model subsume the work done in Milestone 1.b. The array characterizations done in Milestone 1.b were implemented in device specific ways inside the ArrayHandle classes, which is the basis for the data model. These include texture memory support for CUDA and the Array of Structures / Structure of Arrays for the TBB execution environments.

Milestone 2.h Flyweight In Situ-VTK-m Integration (Year 3–Kitware) Fully integrate VTK-m execution and data models (from Milestones 1.a, 2.a, and 2.d) with flyweight in situ (from Milestone 2.c) and demonstrate its application.
Expected Completion: FY17, Q4 Status: Complete

VTK-m has been integrated into Sensei. We have developed a contour analysis module in Sensei and have demonstrated its functionality with the mini apps included in Sensei. Additional filters can now be exposed in Sensei by developing simple analysis modules. All of our work has been included in the upstream Sensei repository.

Milestone 2.i Fault Tolerant Primitive Functions (Year 3–LANL) Develop fault tolerant versions of key data-parallel primitives (such as scan, transform, reduce, etc.) used by VTK-m algorithms in order to transparently provide a level of resiliency within in situ applications.
Expected Completion: FY17, Q4 Status: In Progress

Initial prototype was implemented to catch bad_alloc errors and retry with the streaming array handle. This code has been integrated into the Resilient branch of the vtk-m repository. In pursuing this initial prototype, we found that the scope of this milestone was much broader than originally described, and would require more extensive R&D than was originally scoped. Thus, we focused our research efforts on the remaining milestones.

2.3 Usability

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: Complete

In the area of combustion, we have been working with Dr. Jackie Chen's research group at Sandia National Labs. We have been developing analysis and visualization tools to streamline their current workflow and solve new analysis problems as identified by the domain scientists. We have completed the development of a scalable histogram-based particle selection scheme which couples both in situ and post processing analyses.

Spatially distributed histograms (probability distribution functions) are generated in situ using a set of modules developed at UC Davis. These probability distributions are generated using the full resolution simulation data and can be leveraged in post processing to aid certain analysis tasks. We have been working on investigating how we can use these modules to best interface with the S3D combustion simulation in an unobtrusive manner. Not only must these modules match the scalability of the simulation, scientists need to be able to easily make modifications to the code based on their most recent needs. We have completed scalability tests in large scale production runs.

On the post processing side of things, we have developed a comprehensive visualization software that can utilize the in situ generated histograms to make real time particle selections from large scale datasets. Users can select spatial regions based on desired distributions in the histograms which in turn extracts particle subsets (which were spatially sorted in situ) for later analysis. We have been making detailed design decisions based on a close collaboration with the domain scientists and their analysis needs. We have designed this software (including UI, data management, etc.) so that it can be easily be used by the scientists with little to no modification to their usual workflow.

Our paper describing the recent completed stages of this work was presented at the IEEE LDAV 2016 symposium.

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 at PacificVis 2016.

In this period, we have improved the scalability of halo tag processing. Halo tags are critical for interactive analysis of halo structure, but we must first associate halo tags with their particles in the 3D spatial domain. We have also improved the handling of temporal data in order to facilitate both command line (server-side) and GUI (client-side) use, and have integrated a phase-space mesh rendering method. The first deployment of this interactive visualization facility at Argonne has been completed and is ready for subsequent usability studies to be carried out with full participation of the scientists.

Milestone 3.d Continue Usability Studies (Year 3–UC Davis, U Oregon) Continue studies of Milestone 3.c.
Expected Completion: FY17, Q4 Status: Complete

In supporting the combustion simulations, we have completed initial usability tests for the in situ modules. While our own tests have been successful in showing the scalability of our tools. More evaluation was needed to test the ease at which the domain scientists can invoke or modify our routines as well as their ability to handle the large variety of chemical cases that the S3D simulation can produce. Feedback from the scientists have identified certain key functionalities that would improve usefulness of the software, such as adding the ability to dynamically modify distribution generation parameters while the simulation is in progress. These new components have been added to the in situ modules, which are now fully deployed and integrated within the S3D simulation.

We have also worked on improving the post processing visualization software based on feedback from the domain scientists. We have completed initial usability tests with our domain collaborators to evaluate the ease at which they can use the software and how successful it is in identifying desired trends in the data. We have identified and carefully documented requested improvements to the user interface, interaction techniques, visual depictions of the results, and analysis functionalities. We have also begun implementing user interaction logging into the visualization software in order to generate quantitative measures of usability.

For the cosmology application, we have been working on improving the functionality of the recently deployed interactive visualization facility. Recent trips to Argonne National Laboratory to meet with domain experts have provided additional insight into how scientists can effectively use these new tools. We have identified key areas of improvement, which focus on the interface design and the ways users can interact with each of the data views, and have been modifying our software accordingly.

Milestone 3.e Apply Usability Studies (Year 3–UC Davis, U Oregon) Conduct a review session with each scientist team on usability study results from Milestone 3.d. Refine the design and implementation of techniques and workflow according to lessons learned.

Expected Completion: FY17, Q4 Status: Complete

In our previous work for this milestone, we interviewed stakeholders from the XGC team about their data needs, and documented the findings in a workshop publication. One key finding from the study was how this team is forced to decide which data to keep and which data to throw away.

This has led to us participating in a new algorithm for this team. Specifically, XGC produces significant numbers of particles, each with velocity data. The number of particles is larger than could/should be stored to disk, and so we investigated a method for binning the particles onto a mesh. An important thrust to this research was determining tradeoffs between mesh size, number of particles considered, and the integrity of analyses using the binning (rather than the original data).

We are evaluating these tradeoffs, and submitted a paper to LDAV17. The work was not accepted, but reviewed generally well and we will resubmit it. Further, we anticipate this code ultimately being integrated into ADIOS, which will enable the stakeholders to use it as part of their regular workflow.

For the combustion application, we have applied additional improvements to the post processing visualization software based on feedback from the scientists and our usability tests from the previous milestones. The continuous use of our analysis and visualization tools have validated the effectiveness of using in situ generated distribution functions to the scientists. This has now motivated them to request new analysis tools to explore the statistical properties found within the distributions in more detail. While the current tools will continue to be improved in an iterative fashion, these new tools with become a key part of our continued collaboration with this scientific team.

For the cosmology application, we have also been working on additional improvements to the interactive visualization and analysis tools. Recent evaluations of the deployed system's performance, as well as discussions with domain experts, are allowing us to target specific end-user hardware and architecture types to improve the efficiency of our analysis tools. Furthermore, we have received new requests to facilitate communication and data flow between our tools and other software packages that are commonly used by the domain scientists. Our future collaboration with this scientific team will continue to improve the way our tools can seamlessly fit into their normal analysis workflow.

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 first Mantevo mini-app we are working on is a parallel rendering mini-app, miniGraphics. More specifically, miniGraphics demonstrates sort-last parallel rendering, which is the most common in HPC applications. A sort-last parallel rendering process can be split into two parts: a rendering part that happens locally on a node using data available on that node, and an image compositing part that takes the locally generated images and blends them together. The framework for miniGraphics provides separate implementations for the rendering and the image compositing functions so that they may be mixed to measure different rendering effects. The initial work includes simple reference implementations for both. Future work includes making example rendering that employs OpenGL/GLFW and example compositing using IceT. These implementations provide more industry-standard implementations of rendering and compositing.

The second Mantevo mini-app we are working on is a contouring mini-app, miniIsosurface. Contouring is a basic operation in scientific visualization, has many algorithm variants, and is representative of many interesting visualization algorithms. We have modified and added functionality to the original Marching Cubes mini-app that was implemented in FY15, Q4 to be more general. Our modifications include a reworking of this original implementation to improve clarity of the code and guarantee correctness while still utilizing full compute resources. miniIsosurface now includes additional parallel implementations of the marching cube algorithm and currently have one serial, three OpenMP, one MPI, and one Kokkos-based implementations. In additional to the Marching Cubes algorithm, we have implemented the Flying Edges algorithm, which tends to outperform the Marching Cubes algorithm. We have developed Flying Edges implementations that runs in serial, runs using OpenMP, and runs on the NVIDIA GPU (using the Thrust library).

We are in the process of applying for a copyright assertion for each of the mini-apps. When complete, the mini-app code will be released as part of the Mantevo project (http://mantevo.org).

We have also started using the Ascent in situ visualization framework. Ascent is a lightweight framework which is part of the Alpine project, and is designed to help explore the in situ analysis and visualization needs of simulation code teams running on supercomputers. Ascent contains a thin data model that allows a simulation code to provide the semantics for data and then wrap the data in the VTK-m data model. Ascent also ships with several different simulation mini-apps. We have implemented an isosurface functionality in Ascent, and run several of the simulation mini-apps as test cases. We have also integrated the ADIOS middleware into the Ascent framework which allows us to experiment with both in situ, and in transit scenarios. Using the ADIOS middleware, the mini-apps in can generate self-describing data using the ADIOS Visualization Schema. These data, along with operations to be performed on the data, are written with the ADIOS middleware. We have created several simple visualization services that are based on VTK-m that can ingest the data from the ADIOS stream, and perform the specified operations on the data. These experiments have been performed for in situ, in transit and post-hoc scenarios.

We have also started to use the TAU suite of performance tools to profile in situ workflows. While this work has just started, we anticipate gaining an understanding of how best to deploy VTK-m routines in a way that minimizes the impact on running simulations.

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: Complete

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

Progress on this milestone got delayed due to some shifting of staff. But, this quarter we are reengaging folks from the Oxbow project to continue work on this milestone.

We have performed subsequent analysis using tools from the Oxbow suite to compare the marching cubes implementation in both VTK and VTK-m. As shown in the table below, the instruction mix for both the VTK-m and VTK implementations of the marching cubes algorithms are roughly the same.

	Instruction Mix (%)				
Application	BrOps	IntOps	FpOps	MemOps	Moves
VTK	15.0	35.5	.3	35.8	13.2

VTK-m	12.9	33.9	.2	38.6	14.1

We have also engaged with the TAU performance tools group at the University of Oregon to study the usage of their suite of tools for performance analysis. The TAU suite of tools has the advantage of a lower-barrier of entry and the ability to obtain performance information from both CPU and GPU codes. We have created some test applications for performance analysis that include volume rendering, ray casting, marching cubes, streamlines, and external face extraction. Using TAU, we were able to locate a bottleneck in the marching cubes algorithm during a normal calculation step. The insights into the performance of the code using TAU have enabled the VTK-m team to identify a work around for this performance bottleneck.

4.c Architectural Studies (Year 3–SNL, UC Davis, ORNL) Use the mini-applications, simulation, and experiments to perform studies of behavior on future architectures, including performance, memory hierarchy usage, and heterogeneous component use.

Expected Completion: FY17, Q4 Status: Complete

We are currently studying the performance of the particle advection filter functionality on a variety of different architectures. The hardware we are targeting include the Titan supercomputer at ORNL, the Rhea analysis cluster at ORNL, and the test bed for Summit at ORNL. We are targeting both CPU and GPU implementations, as well as studying the scalability as a function of number of cores on CPU implementations. We are testing both a particle advection algorithm, which will be used for applications like FTLE, as well as a streamline algorithm where particle paths are recorded.

One focus of our studies is the portability of particle advection algorithms in VTKm, and how they compare to hand-tuned implementations for both CPU and GPU. In order to study the portability, we have run a large series of tests on the VTKm implementation (for both CPU and GPU), and a previously published hardware specific implementation (for both CPU and GPU). The tests that we have run include a wide variety of workloads, including both numbers of particles, duration of advection, and types of vector fields. The two tables below summarizes our initial findings. Each workload, denoted W_i, is run on a number of different GPUs. The **File** column denotes the type of vector field, the **CUDA Cod**e columns denote the timings for the hardware-specific particle advection implementation, and the **VTK-m Comparision** columns list the speed-up factors for the VTK-m implementation. A factor of > 1X indicates that VTK-m is faster.

	File	CUDA Code K20X	K80	P100	VTK-m Comparison K20X	K80	P100
W_1	Astro	0.844s	0.285s	0.836s	1.35X	0.55Xs	2.15X
	Fusion	0.845s	0.284s	0.838s	1.35X	0.55X	2.17X
	Thermal	0.844s	0.284s	0.837s	1.35X	0.54X	2.14X
W_2	Astro	0.869s	0.301s	0.842s	1.34X	0.55X	2.08X
	Fusion	0.874s	0.304s	0.845s	1.35X	0.56X	2.11X
	Thermal	0.871s	0.304s	0.844s	1.34X	0.56X	2.14X
W_3	Astro	3.418s	1.959s	2.353s	2.26X	2.07X	4.08X
	Fusion	3.367s	1.824s	2.219s	2.23X	1.90X	3.81X
	Thermal	3.327s	1.856s	2.247s	2.21X	1.96X	3.85X
W_4	Astro	6.682s	5.067s	3.564s	1.29X	1.78X	2.02X
	Fusion	8.803s	6.763s	4.420s	1.65X	2.42X	2.49X
	Thermal	8.793s	6.830s	4.500s	1.72X	2.45X	2.53X
W_5	Astro	14.353s	12.963s	6.464s	0.37X	0.56X	0.48X
	Fusion	63.993s	54.670s	25.694s	1.56X	2.24X	1.88X
	Thermal	56.133s	49.172s	23.161s	1.42X	2.04X	1.70X

3 Other Activities

3.1 Publications

"Scalable Visualization of Time-varying Multi-parameter Distributions Using Spatially Organized Histograms." T. Neuroth, F. Sauer, W. Wang, S. Ethier, C. S. Chang and K. L. Ma. *IEEE Transactions on Visualization and Computer Graphics*, (preprint).

"In situ Visualization in Radiation Transport using VTK-m." Mark Kim, Tom Evans, Scott Klasky, David Pugmire. *Workshop on In Situ Analysis and Visualization (ISAV)*, November 2017 (to appear).

"A Combined Eulerian-Lagrangian Data Representation for Large-Scale Applications." F. Sauer, J. Xie, and K. L. Ma. *IEEE Transactions on Visualization and Computer Graphics*, 23(10), 2248-2261, October 2017.

"Maximal Clique Enumeration with Data-Parallel Primitives." Brenton Lessley, Talita Perciano, Manish Mathai, Hank Childs, and E. Wes Bethel. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2017.

"Techniques for Data-Parallel Searching for Duplicate Elements." Brenton Lessley, Kenneth Moreland, Matthew Larsen, and Hank Childs. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2017.

"Spatio-Temporal Feature Exploration in Combined Particle/Volume Reference Frames." F. Sauer and K. L. Ma. *IEEE Transactions on Visualization and Computer Graphics*, 23(6), 1624-1635, June 2017.

"Achieving Portable Performance for Wavelet Compression Using Data Parallel Primitives." Shaomeng Li, Nicole Marsaglia, Vincent Chen, Christopher Sewell, John Clyne, and Hank Childs. In *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, June 2017.

"Preparing for In Situ Processing on Upcoming Leading-edge Supercomputers", J. Kress, R. Churchill, S. Klasky, M. Kim, H.Childs, D. Pugmire, *Supercomputing Frontiers and Innovations*, December, 2016.

"Audience-Targeted Design Considerations for Effective Scientific Storytelling." F. Sauer, T. Neuroth, J. Chu and K. L. Ma. *Computing in Science & Engineering*, 18(6), 68-76, November/December 2016.

"Visualization and Analysis Requirements for In Situ Processing for a Large-Scale Fusion Simulation Code." James Kress, David Pugmire, Scott Klasky, and Hank Childs. In *SC16 Workshop on In Situ Analysis and Visualization (ISAV)*, November 2016.

"Optimizing Multi-Image Sort-Last Parallel Rendering." Matthew Larsen, Kenneth Moreland, Chris Johnson, and Hank Childs. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2016.

"Parallel Peak Pruning for Scalable SMP Contour Tree Computation." Hamish Carr, Gunther Weber, Christopher Sewell, and James Ahrens. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2016. Best Paper Award.

"In Situ Generated Probability Distribution Functions for Interactive Post Hoc Visualization and Analysis." Yucong Ye, Tyson Neuroth, Franz Sauer, Kwan-Liu Ma, Giulio Borghesi, Aditya Konduri, Hemanth Kolla, and Jacqueline Chen. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2016.

"Parallel Distributed, GPU-Accelerated, Advanced Lighting Calculations for Large-Scale Volume Visualization." Min Shih, Silvio Rizzi, Joseph Insley, Thomas Uram, Venkatram Vishwanath, Mark Hereld, Michael E. Papka, and Kwan-Liu Ma. In *Proceedings of the IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, October 2016.

"Hybrid Data-Parallel Contour Tree Computation." Hamish Carr, Christopher Sewell, Li-Ta Lo, and James Ahrens. In *Proceedings of the Computer Graphics and Visual Computing Conference*, September 2016.

"External Facelist Calculation with Data-Parallel Primitives." Brenton Lessley, Roba Binyahib, Robert Maynard, and Hank Childs. In *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, June 2016.

"VTK-m: Accelerating the Visualization Toolkit for Massively Threaded Architectures." Kenneth Moreland, Christopher Sewell, William Usher, Li-ta Lo, Jeremy Meredith, David Pugmire, James Kress, Hendrik Schroots, Kwan-Liu Ma, Hank Childs, Matthew Larsen, Chun-Ming Chen, Robert Maynard, and Berk Geveci. *IEEE Computer Graphics and Applications*, 36(3), May/June 2016.

"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 Plugins using VTKm for In-Transit Visualization with ADIOS." D. Pugmire, J. Kress, J. Meredith, H. Childs, M. Larsen, S. Klasky, J. Choi, and N. Podhorszki. In *Supercomputing Frontiers*, Singapore, March. 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

Tutorials Chair, *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Hank Childs, November 12–17, 2017.

Papers Co-Chair, *In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV)*, Kenneth Moreland, November 12, 2017.

Papers Co-Chair, *Large Scale Data Analysis and Visualization (LDAV)*, Kenneth Moreland, October 2, 2017.

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

Papers Chair, *International Symposium on Graph Drawing & Network Visualization (GD)*, Kwan-Liu Ma, September 25–27, 2017.

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, *IEEE Big Data*, Kwan-Liu Ma, December 11–14, 2017.

Program Committee, *ACM SIGGRAPH ASIA Symposium on Visualization (SOV17)*, Kenneth Moreland, November 27–30, 2017.

Best Papers Committee, *ACM SIGGRAPH ASIA Symposium on Visualization (SOV17)*, Kwan-Liu Ma, November 27–30, 2017.

Program Committee, *Visualization Showcase SC17*, Kenneth Moreland, November 12–17, 2017.

Papers Co-Chair, *In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV)*, Kenneth Moreland, November 12, 2017.

Program Committee, *Pacific Graphics*, Kwan-Liu Ma, October 16-19, 2017.

Program Committee, *IEEE Scientific Visualization (SciVis)*, David Pugmire, October 1-6, 2017.

Program Committee, *IEEE Scientific Visualization (SciVis)*, Hank Childs, October 1-6, 2017.

Best Papers Committee, *IEEE Scientific Visualization (SciVis)*, Kwan-Liu Ma, October 1-6, 2017.

Program Committee, *IEEE Conference on Visual Analytics Science and Technology (VAST)*, Kwan-Liu Ma, October 1-6, 2017.

Best Papers Committee, *IEEE Conference on Visual Analytics Science and Technology (VAST)*, Kwan-Liu Ma, October 1-6, 2017.

Steering Committee, *Large Scale Data Analysis and Visualization (LDAV)*, Kwan-Liu Ma, October 2, 2017.

Program Committee, *Large Scale Data Analysis and Visualization (LDAV)*, Hank Childs, October 2, 2017.

Steering Committee, *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Kwan-Liu Ma, October 2, 2017.

Program Committee, *IEEE Conference on Software Visualization (VISSOFT)*, Kwan-Liu Ma, September 18–19, 2017.

Program Committee, *IEEE Cluster 2017*, David Pugmire, September 5-8, 2017.

Program Committee, *IEEE Cluster 2017*, Hank Childs, September 5-8, 2017.

Program Committee, *EG/VGTC Conference on Visualization (EuroVis)*, Kenneth Moreland, June 12–16, 2017.

Program Committee, *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, Kenneth Moreland, June 12–13.

Steering Committee, *IEEE PacificVis*, Kwan-Liu Ma, April 18–21, 2017.

Program Committee, *SPIE Visualization and Data Analysis (VDA)*, Hank Childs, January 29–February 2, 2017.

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

Program Committee, *ACM SIGGRAPH ASIA 2016 Symposium on Visualization (SA16VIS)*, Kwan-Liu Ma, December 5–8, 2016.

Program Committee, *In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV)*, Kenneth Moreland, November 13, 2016.

Program Committee, *In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV)*, Hank Childs, November 13, 2016.

Program Committee, *Visualization Showcase SC16*, David Pugmire, November 2016.

Program Committee, *Cooperative Design, Visualization, and Engineering (CDVE)*, Kwan-Liu Ma, October 24–27.

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

Program Committee, *IEEE Symposium on Visualization for Cybersecurity (VizSec)*, Kwan-Liu Ma, October 24, 2016.

Steering Committee, *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Kwan-Liu Ma, October 24, 2016.

Program Committee, *IEEE Large Data Analysis and Visualization (LDAV)*, Christopher Sewell, October 23, 2016.

Program Committee, *IEEE Large Data Analysis and Visualization (LDAV)*, Kwan-Liu Ma, October 23, 2016.

Program Committee, *Graph Drawing & Network Visualization*, Kwan-Liu Ma, September 19–21, 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, *IEEE Conference on Multimedia and Big Data (BigMM)*, Kwan-Liu Ma, April 20–22, 2016.

Program Committee, *IEEE BigDataService*, Kwan-Liu Ma, March 29–April 1, 2016.

Program Committee, *Workshop on Emotion and Visualization (EmoVis)*, Kwan-Liu Ma, March 10, 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.

Steering Committee, *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Kwan-Liu Ma, October 26, 2017.

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

"Big Data Visualization," Keynote Speaker, Kwan-Liu Ma, *Taiwan Data Science Conference*, Taipei, Taiwan, November 11, 2017.

"Visualization for Scientific Discovery and Storytelling," Invited talk, Kwan-Liu Ma, Oak Ridge National Laboratory, Tennessee, August 24, 2017.

"In Situ Processing: Opportunities, Challenges, and Instantiations," Hank Childs, *ISC High Performance Conference*, Frankfurt, Germany, June 2017.

"State of the Art for In Situ Visualization," Keynote, Hank Childs, *IXPUG Workshop on Software-Defined Visualization*, Austin, TX, May 2017.

"Audience Targeted Exploratory and Explanatory Visualization," Keynote Speaker, Kwan-Liu Ma, *Spring Conference on Computer Graphics (SCCG)*, Czech Republic, May 15-17, 2017.

"In Transit Visualization of Full and Reduced Simulation Data on HPC Platforms", Invited Talk, David Pugmire, *Parallel CFD Conference*, Glasgow Scotland, May 2017.

"Emerging Topics in Visual Analytics," Keynote, Kwan-Liu Ma, *PacificVAST*, Seoul, South Korea, April 18, 2017.

"Data Visualization," Invited lecture, Kwan-Liu Ma, UC Davis Medical School, Sacramento, CA, February 8, 2017.

"Visualization for the Public," Invited talk, Kwan-Liu Ma, University of Tokyo, Tokyo, Japan, December 19, 2016.

"Big Data Visualization," Invited Talk, Kwan-Liu Ma, Fuji Xerox, Yokohama, Japan, December 15, 2016.

"Introduction to Data Visualization," Invited Lecture, Kwan-Liu Ma, Keio University, Yokohama, Japan, December 13, 2016.

"Audience-Targeted Visualization Designs for Effective Storytelling," Invited talk, Kwan-Liu Ma, Keio University, Yokohama, Japan, December 8, 2016.

"Visualization for You," Keynote Speech, Kwan-Liu Ma, *Australian Conference on Human-Computer Interaction (OzCHI)*, Tasmania, Australia, November 30, 2016.

"Exascale Visualization and In Situ Processing," Invited lecture, Hank Childs, University of Tennessee, Knoxville, TN, September 2016.

"In Situ Processing: Opportunities, Challenges, and Instantiations," Invited talk, Hank Childs, *Smoky Mountains Computational Sciences and Engineering Conference*, September 1, 2016.

"In Situ Processing: Opportunities, Challenges, and Instantiations," Invited talk, Oak Ridge National Laboratory, Oak Ridge, TN, August 2016.

"Visualization: A Tool for Data Exploration and Storytelling," Invited lecture, Kwan-Liu Ma, Hokudai University, Hokkaido, Japan, August 9, 2016.

"Visualization and Analysis Services for Extreme Scale Computing" Invited lecture, David Pugmire, Rutherford Appleton Laboratory, UK, July 2016

"Visualization and Analysis Services for Extreme Scale Computing"**,** Invited lecture, David Pugmire**,** Oxford University, UK, July 2016

"Visualization and Analysis Services for Extreme Scale Computing"**,** Invited lecture, David Pugmire, Swansea University UK, July 2016

"Visualization and Analysis Services for Extreme Scale Computing" Invited lecture David Pugmire**,** Daresbury Laboratory, UK, July 2016

"Visualization and Analysis Services for Extreme Scale Computing" Invited lecture David Pugmire**,** University of Leeds, UK July 2016

"Scientific Visualization for the Public," Invited talk, Kwan-Liu Ma, Hanzhou Low Carbon Science & Technology Museum, Hangzhou, China, July 8, 2016.

"Visualization: An Exploratory and Explanatory Tool," Invited talk, Kwan-Liu Ma. *International Symposium on Visual Computing*, Hangzhou, China, July 7, 2016.

"Recent Advances in Visualization Research," Invited seminar, Kwan-Liu Ma, National Chiao Tung University, Hsinchu, Taiwan, June 29, 2016.

"Big-Data Visualization Techniques for Studying Behaviors, Connections, and Evolution," Invited Talk, Kwan-Liu Ma, Institute of Statistical Science, Academia Sinica, Taipei, Taiwan, June 17, 2016.

"Visualization: An Essential Tool for Scientific Discovery and Storytelling," Invited talk, Kwan-Liu Ma, Pacific Science Congress, Academia Sinica, Taipei, Taiwan, June 16, 2016.

"Visualization: A Tool for Exploration and Storytelling," Invited talk, Kwan-Liu Ma, Biophotonics Seminar, UC Davis, June 2, 2016.

"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.

"Visualization Toolkit: Improving Rendering and Compute on GPUs," Presentation, Robert Maynard, *GPU Technology Conference*, April 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.

"Exascale Visualization: What Will Change," Invited talk, Hank Childs, National Center for Atmospheric Research, Boulder, CO, March 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.

"VTK-m: Building a Visualization Toolkit for Massively Threaded Architectures," Invited presentation, *Ultrascale Visualization Workshop*, November 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 multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND 2017-5709 R