National Visual Analytics Center (NVAC) Workshop
PNNL, October 28, 2004

1 Executive Summary

During the period October 28-29 2004, a panel of representatives from several DOE National Laboratories convened at Pacific Northwest National Laboratory (PNNL) in Richland, WA, to be briefed on the National Visualization Analytics Center's (NVAC) current Functional and Operational Architecture design. This document contains the panel's responses to the presentation from and the discussion with representatives from the PNNL NVAC architecture team.

The responses are organized into comments about the NVAC architecture, comments about interoperability in general and NVAC interoperability in particular, and comments about Intellectual Property (IP). The panel also provided – upon request – a vignette of technologies that several National Laboratories (strategic and non-strategic) could contribute to the NVAC effort.

With regard to the NVAC architecture, the panel observed that the breadth and depth of the NVAC architecture is quite large, which is not unexpected given the intended scope of the NVAC project. The panel provided numerous suggestions about how NVAC might further clarify the operational and functional objectives of the architecture, and suggested a divide-and-conquer strategy to help limit scope in order to achieve success.

The panel observed that previous efforts within DOE aimed at addressing software and data interoperability have produced mixed results. Having learned from these experiences, the panel made suggestions aimed at refining the NVAC architecture as well as fostering an open, sharing environment that will be viewed as inclusive rather than exclusive within the community of NVAC participants.

2 NVAC Architecture: Comments and Suggestions

Use-Case Scenarios. It is difficult to do a thorough evaluation of the NVAC architecture without well-defined statements about how the architecture will be used. For example, a set of use scenarios (expressions of how various end users might use the system) would be helpful in establishing requirements in terms of capability and capacity. The scenarios should embody reasonable performance and capability metrics, including hardware, connectivity, multiple and heterogeneous data sources, variety of simulations, data rates, information flows, supporting multiple analysts, R&D personnel, field personnel, and communications. The scenarios should reflect current and anticipated needs, and will be helpful in defining the functional and operational capability and capacity limits of the architecture as well as overall architecture requirements.

Incremental Design and Implementation. The most important aspect of the NVAC Architecture design is limiting the scope of the integration effort to be undertaken. Given the potential for so many different software systems, libraries and services, in a variety of functional categories, it will be important to choose an incremental design approach, both with respect to the number and type of systems to incorporate and in the rigor with which the myriad of necessary software interfaces will be developed.

Define Architecture Requirements. The materials reviewed by the panel (presentations, architectural diagrams) focused on how the NVAC may be structured. While such information shows a good start, there were no explicit statements about NVAC's functional or operational requirements. The basic hard and fast requirements for NVAC need to be defined and agreed upon in order to help set boundaries on what can (and can not) be reasonably accomplished.

Interface Specification. With an emphasis on long-term interoperability (see Section Three), the NVAC Architecture must be assembled piece by piece, with carefully selected candidates for the initial round of tools and technology. The software interfaces must be hierarchical in terms of scope. That is, at every level in the architecture, a strict interface must be in place to define the methods of interaction between any two components. This is true at a high level, where services interact with each other, applications, and with lower level libraries. It is also true at a lower level, where libraries will interact with each other as well as lower level libraries.

By defining this strict hierarchical set of interfaces, the need for a broad, all-encompassing data model will not be necessary. Components of the architecture can be developed in a flexible, modular, reusable fashion, and only interface changes will have an impact on the overall use of the architecture. It is then possible to change data models and/or technologies without affecting the entire NVAC architecture and its users. Thus this approach will avoid failings of past projects where "the one true data model" has failed to emerge despite concerted effort

Data Adapters. The architecture does not require an explicit "NVAC common, all-encompassing data model." Instead, the architecture calls for a metadata model description to flexibly accommodate a number of community data models. The metadata model approach will facilitate growth and adaptation over time as requirements evolve. To realize its use in practice, the metadata model approach will likely require an "adapter layer" to marshal or present data to functional software components or tools that are designed and implemented with a static interface.

Include Security Throughout Design. Security, privacy, data provenance, access rights, and so forth should be accessible to all software components throughout the call stack. Similarly, digital rights management, if of concern to NVAC, should be included in the operational and functional architecture. A well-defined security model must be a part of the initial design, and must be a part of the entire functional and operational architecture. Security should not be "shoehorned" into an existing design – it must be integral to the design.

Other Specific Comments. The word "security" should be added to the Crosscutting column of the Functional Architecture. Journaling / Logging/ Problem Solving Environment should also be added. A question to ponder is "Who are users of the NVAC functional architecture?" The NVAC architects might use "comparative visualization" instead of "multi-visualization techniques" and "Rendering" instead of "presentation".

3 NVAC Interoperability

3.1 Interoperability Definitions and Challenges

Data and Software Interoperability. The panel envisions two broad areas of interoperability from a systems architecture perspective: data and software. Data interoperability means that the output of a software tool can be used as input to another without format conversion. Software interoperability is broadly interpreted to mean consistent interfaces, where the interface definition depends upon access method: there are compile-time interfaces (subroutines/function parameter lists) and dynamic interfaces (services, protocols).

Webster's provides two entries for the term *interoperability* that embody both broad areas:

- The ability to exchange and use information.
- The ability of software and hardware on multiple machines from multiple vendors to communicate.

The basic definitions can be extended to include the notion that there is a dimension of intra- and inter-institution sharing. This added dimension helps provide the distinction between "theoretical" and "practical" interoperability.

Primary Benefit of Interoperability. Interoperability provides the flexibility to more quickly create domain-specific solutions that would otherwise be possible only through separate, concurrent development efforts. It helps to eliminate expensive duplication of effort where individual projects each tend to reinvent crucial infrastructure components.

Funding Interoperability. The panel observes that the traditional funding model for visualization research and development has tended to focus on "point solutions" that result in a journal article and a research prototype implementation. Such an approach does not foster an emphasis upon infrastructure and interoperability. Interoperability and interoperable infrastructure need to be acknowledged as a priority objective for the visualization and data analysis community and funded as such. The NVAC project seems to be taking this approach, which is viewed as a positive step in the right direction.

3.2 Interoperability Concerns and Suggestions

The NVAC scope as described in presentations and documents is quite broad and deep. Its breadth is reflected by the large number of potential capabilities present at each level in the "Functional Architecture" diagram. Its depth is reflected by the functional scope that captures the entire visual analysis pipeline from data bytes to onscreen pixels.

As described and presented to us, the NVAC scope is very ambitious. To help ensure success of the project, the panel suggests a concerted effort where appropriate use scenarios are clearly defined. The use scenarios should include descriptions of the actual intended domain specific features and applications built within the NVAC architecture. Further, these scenarios should vary in terms of their operational requirements. For example, one scenario might dictate that a decision be reached within a specified period of time (a few minutes or hours). Another may call for a comparison of conclusions drawn using analyses from different datasets. The importance of these use scenarios cannot be overstressed.

The panel suggests that NVAC define a spectrum of use scenarios that:

- Help define the complete scope of operational and functional architecture requirements, including interfaces between and within architectural layers;
- Can serve as a vehicle for defining project priorities so that the architecture and its development may occur in a scope-limited and incremental fashion;
- Help explain the project and its goals to stakeholders and other interested parties;
- Include both near- and far-term scenarios, as well as routine and extraordinary applications (e.g., NVAC is asked to deliver answers in a small window of time for an urgent, time-critical incident).

The panel expects that insights into the commonalities of interface requirements, data models, execution control, security requirements, use of multiple/distributed resources, etc., will emerge as part of the analysis of these diverse use case scenarios. Additionally, the panel expects that the use scenarios will serve as the basis for a clear set of definitions for operational and functional requirements.

4 Intellectual Property Issues

The panel engaged in discussion about the subject of IP, with particular emphasis upon the issue of using an Open Source license as the basis for software releases.

By way of background, in the summer of 2003, the Department of Energy (DOE)'s Office of Advanced Scientific Computing Research (OASCR) issued a memo stating that "all publicly released DOE Laboratory software funded by Advanced Simulation and Computing (ASCI) or OASCR developed by the National Laboratories shall be released using an appropriate Open Source license; headquarters approval is required for a more restrictive license either for commercialization or for export control (security) reasons." The memo listed several reasons favoring Open Source licenses, and the items included the usual arguments: a "hedge" against change in support status over time for "mission critical" software; increased reliability and decreased security exposure as flaws are quickly identified and repaired; better leverage of government investments; contributions to state of the art, and so forth.

As NVAC moves forward, the panel suggests adopting a similar position with respect to the Open Source issue: all software should be released using an appropriate Open Source license unless there is a compelling reason not to do so. Such a posture and practice will have broad, far-reaching benefits for all participants:

- For interface/protocol definitions, an Open Source reference implementation fosters "many players," each of which may use a different IP model. This approach will be friendly to commercial and research concerns alike.
- An Open Source reference implementation of key NVAC infrastructure components will provide a tangible and working set of capabilities for the RVAC participants, saving them the trouble of each designing and implementing critical infrastructure.
- A clearly defined Open Source release policy will mitigate future concerns about IP, and help streamline the process for inter-institution collaboration.
- Open Source reference implementations of key NVAC infrastructure will help promote interoperability, which is a central NVAC tenet.

While the general NVAC IP posture could be one that promotes Open Source software releases, careful "control" should be exercised to ensure the integrity of source code. In other words, the practice of Open Source need not be a "free for all" where anyone can check in any code changes to the central CVS tree. Instead, the NVAC project management process should include a rigorous set of quality controls and software engineering practices (e.g., unit tests, regression tests) that ensure the integrity and correctness of all contributed and internally developed source code.

Some on the panel suggested that an Open Source approach might not be attractive to external participants due to the perceived "risk" that an institution's individual contributions would not be sufficiently well recognized. While this is a valid concern, there is an approach that helps ensure that the identity of each individual institution's contributions would remain intact: each institution can reserve the right to assert Copyright on their published work (of software), and simply grant non-exclusive use rights to anyone who wants to use the work. This approach is essentially a BSD-style license, and is the approach currently in use at many of the DOE laboratories.

In summary, the panel believes that NVAC and its related programs will benefit greatly from a position that uses an Open Source license as the primary vehicle for managing Intellectual Property concerns and software releases. The benefits will be realized in many ways, including technical, sociological, economic and long-term project vigor. The Open Source license must be chosen carefully so as to be friendly to industry, academia and government alike. At the same time, it is not the case that all technologies and applications that are developed under the NVAC umbrella need be released as Open Source, and there are very good reasons not to do so in many cases.

5 National Labs' Contributions to the NVAC

The following sections outline a set of ongoing projects and core competencies at DOE National Laboratories that have relevance to the NVAC mission. The material is organized into five main categories: architecture and design, scalability, performance, functionality, and more general research/development. The following list is in no way exhaustive nor complete. It simply represents an example cross-section of DOE competencies the panel feels are applicable to NVAC – a complete and exhaustive list would fill many pages, and is beyond the scope of this document.

5.1 Architecture and Design

Similar Framework Efforts. The Distributed Visualization Architecture (DiVA) project[1] aims to define a set of specifications and technologies that foster interoperability in remote, distributed and high-performance visualization contexts. With interoperability as one of its central tenants, DiVA and NVAC share a common vision. LBNL is the lead institution on the project, and several other institutions have participated at varying levels in design/requirements meetings (ORNL, LLNL). Having some of the DiVA architects on the NVAC "NEAT" architectural design/review team would be of benefit to both projects.

Component Architectures and Component Interfaces. DOE's SciDAC Common Component Architecture[2] (CCA) project aims to define a methodology for defining and implementing software components and associated interfaces, with high performance computing as its primary target. The CCA effort includes a number of subprojects that have some relevance to NVAC architectural design and implementation. The Babel[3] project from LLNL provides the ability to generate cross-language interfaces, and supports automatic generation of language interface bindings for C, C++, Fortran 77/90, Java and Python. Babel is the core element of CCA in terms of providing inter-language compatibility.

The general CCA tools and component frameworks could provide the basis for a common software substrate and a standardized component-based implementation paradigm for assembling the various software components and services required to process and visualize information in the NVAC architecture. CCA technology and tools could be applied to form the initial infrastructure of the Visual Analytics framework, with a variety of parallel and distributed run-time control flow options. ORNL has significant experience in applying the emerging CCA technology to new projects as evidenced by their MxN project[4], which implements data communication and redistribution between parallel components.

Interface Specification. A fundamental and crucial hurdle to the NVAC integration effort is the development of interface specifications for communication and interaction among the many software components that will ultimately make up the system. The software components/processes may take many forms, and each form presents a unique set of challenges in terms of interface design.

---

[1] DiVA website: http://vis.lbl.gov/Research/DiVA/
[2] CCA Forum website: http://www.cca-forum.org/
[3] http://www.llnl.gov/CASC/components/babel.html

[4] MxN Project website: http://www.csm.ornl.gov/cca/mxn/

Several groups within DOE have experience designing and implementing complex applications from individual components. ORNL has practical experience guiding the development of functional interface specifications from CCA componentization work with a variety of scientific researchers from the chemistry, climate and fusion domains. Further, in collaboration with LBNL, Utah and other research organizations, ORNL has been exploring preliminary component-based interfaces and solutions for remote and distributed visualization and streaming environments, as part of the DiVA forum. This early effort aligns well with the goals of the NVAC Architecture, and could provide essential insight and infrastructure for the basic technological challenges in the construction of generalized interfaces for scalable data analysis and visualization pipelines.

A different avenue, also in line with NVAC objectives, is the notion of persistent services accessed on a dynamic, on-demand basis. A good deal of work in this area has been pursued by the Open Grid Services Architecture[5] project (OGSA). The OGSA represents a transition to persistent, web-based services (WSDL, Soap) for distributed computational components accessed through a single authentication mechanism. Many institutions have expertise in this area.

5.2 Scalability

Distributed Database Queries. Large distributed databases and data sources present several unique challenges, including efficient and effective analysis, storage and visualization. Traditional approaches would require an expensive gathering step to centralize data for analysis and rendering. One approach that holds promise for distributed processing is the use of “agent-based” methods. Such an approach would reduce the bottleneck of centralized data gathering for visualization and analysis, thereby increasing throughput potential. Several organizations within DOE have a competency in this area (ORNL, LBNL).

Semantic Graph Algorithms. The complexity of data and information sources in the current environment impedes intelligent information analysis and knowledge discovery. The relational property of this data makes graphs a natural mechanism for representing the relationships between objects. Semantic graphs have emerged as a mechanism for imposing structure on relational graphs to make data analysis easier to interpret in context. Processing intelligent queries, e.g. for DHS's ADVISE and Biodefense Knowledge Center, presents a significant computational challenge. Semantic graph algorithms could be applied to: (a) identify a minimum group of people that are related to all the other people of interest (minimum vertex cover); (b) discover a suspicious pattern of interest in the large database (subgraph isomorphism); or (c) find the largest group of cities such that each pair of cities could be affected by a disease spreading from one to another, or enumerate all such groups (maximum or maximal cliques). All these graph problems can be classified as NP-complete or harder, and as such have been viewed (until recently) as computationally intractable. Several DOE organizations have ongoing efforts in these areas (ORNL, LLNL).

---

[5] OGSA website: http://www.globus.org/ogsa/

Distributed Computing. CUMULVS[6], a middleware software technology from ORNL, could provide useful capabilities and protocols for managing collections of large and distributed data sets, especially those originating from live high-performance simulation programs. Using CUMULVS, an interactive data conduit can be attached dynamically to an active simulation program, to provide online visualization of intermediate results, computational steering of any user-defined algorithmic or model parameters, parallel data exchange and redistribution for coupled simulations, and even application-directed checkpointing and automated failure recovery for the simulation environments. CUMULVS handles the complex data collection details for parallel decomposed or distributed data sets.

Many Simultaneous Users. One of the operational objectives made known to the visiting panel was the need to support many simultaneous users. Several different existing DOE competencies will be directly applicable for supporting many simultaneous users (thousands). LBNL's Pervasive Computing project[7] provides technologies for secure group communications and resource-sharing, along with demonstrated expertise in supporting large, distributed teams of scientific researchers (National FusionGrid[8], Particle Physics Data Grid[9]).

5.3 Performance

Streaming and Multiresolution Data. It is generally agreed that streaming data transport and processing paradigms will result in enhanced performance for data-intensive applications. LLNL's REQUEST project[10] is research into methods for the real-time query of sensor and image data. LLNL's Visus adds the notion of streaming and remote queries to ASCI/Views hierarchical indexing methods for large data[11]. LBNL's Bro network monitoring system[12] not only monitors and analyzes network traffic in real time for threats, but also interfaces directly with border routers to control network traffic in response to threats as they evolve.

Scalable, Distributed and Remote Visualization and Analysis. Remote visualization refers to the scenario where data and analyst are in different locations. LBNL's Visapult project[13] implements a pipelined parallel architecture for high-throughput, high performance, latency tolerant visualization. LLNL's VisIt[14] is a general-purpose visualization application built using a pipelined parallel architecture. ParaView[15] is an Open Source parallel visualization application built using the Visualization Toolkit[16]

[6] CUMULVS website: http://www.csm.ornl.gov/cs/cumulvs.html
[7] Pervasive Computing Environment website: http://www-dsd.lbl.gov/Collaboratories/pcce.html
[8] National FusionGrid website: http://www.fusiongrid.org/
[9] Particle Physics Data Grid website: http://www.ppdg.net/
[10] REQUEST website: http://www.llnl.gov/casc/request/
[11] ASCI/Views Novel Data Representations website: http://www.llnl.gov/icc/sdd/img/data.shtml
[12] Bro website: http://www.bro-ids.org/
[13] Visapult website: http://vis.lbl.gov/Research/visapult2/index.html
[14] VisIt website: http://www.llnl.gov/visit/
[15] ParaView website: http://www.paraview.org/HTML/Index.html
[16] VTK website: http://www.vtk.org/

along with technical input and funding from ASCI. Cactus[17] offers a run-time configurable framework for simulation that supports remote visualization and computational steering in the form of a web-based interface. LBNL's Distributed Monitoring Framework[18] emphasizes monitoring and tuning to improve end-to-end throughput in distributed workflows. Sandia's ICE-T[19] is a compositing system that allows a cluster of small machines to drive a large, high-resolution display. Chromium[20] is a general-purpose technology that is widely used for deploying OpenGL-based applications on scalable rendering platforms – from tiled display systems to sort-last compositing/processing configurations – without the need to modify the application.

Alternative Computing and Analysis Platforms. Given additional support, ORNL's existing DHS project for the development of Parallel-R[21] can be leveraged, in collaboration with LANL and LLNL, to port statistical analysis implementations onto GPUs and FPGAs. These efforts represent an optimization leveraging special-purpose computing hardware to provide better throughput and performance for large-scale, data-intensive analysis applications.

5.4 Functionality

Federated databases and queries. Several efforts within DOE focus on providing the capability to perform queries and searches over large, distributed and federated databases. The Earth Systems Grid and Storage Resource Manager projects from LBNL[22] aim to provide highly efficient queries across federated databases, and to use "query brokerages" to insulate applications from the complexity of accessing multiple, distributed databases. The Earth System Modeling Framework[23] is building high-performance, flexible software infrastructure to increase ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, data assimilation, and other Earth science applications. The ESMF defines an architecture for composing multi-component applications and includes data structures and utilities for developing model components. The Data Foundry project[24] at LLNL aims to improve access to distributed, heterogeneous data, for example through multi-database or data warehousing techniques; to reduce the size of the data sets being analyzed, for example by filtering data; to provide novel ways of interacting with the data, for example allowing a broad range of user defined queries; to determine appropriate ways to store data for efficient retrieval.

Finding, Sharing and Using Components and Services. The Alexandria[25] project at LLNL is a research effort in the development, cataloging, and distribution of component-based software. It will be used as the component repository in the Common Component

---

[17] Cactus website: http://www.cactuscode.org/
[18] Distributed Monitoring Framework website: http://www-didc.lbl.gov/DMF/
[19] ICE-T: See "Sort-Last Parallel Rendering for Viewing Extremely Large Data Sets on Tile Displays", Moreland et al, available at http://www.cs.unm.edu/~kmorel/documents/PVG2001.pdf
[20] Chromium website: http://chromium.sourceforge.net/
[21] Parallel-R Large-Scale Statistical Computing: http://www.aspect-sdm.org/Parallel-R/
[22] Earth Systems Grid, Earth Systems Modeling Framework projects: http://sdm.lbl.gov/
[23] Earth System Modeling Framework: http://www.esmf.ucar.edu/
[24] Data Foundry website: http://www.llnl.gov/CASC/datafoundry/
[25] Alexandria website: http://www.llnl.gov/CASC/components/alexandria.html

Architecture infrastructure. Taking a slightly different approach aimed at runtime resource discovery and use, the Grid community's Open Grid Services Architecture[26] defines a framework for defining and locating services in heterogeneous collections of resources. The OGSA is built on standard web services, and provides the ability to locate and use remotely located, persistent services on computational grids.

Graph Layout and Visualization. The National Labs have a diverse portfolio of research and implementation efforts in the general topic area of graph layout and visualization. LANL and LBNL have ongoing efforts that focus on graph layout, partitioning, and visualization primarily in the area of cybersecurity. SNL-NM is deploying in-house tools for concept analysis, graph layout, and graph display using VTK (an open-source visualization toolkit) as a framework. This effort involves adding information visualization capability to VTK. As the interfaces in the NVAC architecture solidify, SNL can write an interface layer to bring VTK quickly to bear as a toolkit for building components and mission platforms.

Application Building Kits. One of the challenges in implementing and using distributed computing applications is the need to encapsulate the complexity of distributed platforms from developers and users. Visual programming environments provide the ability to use a point-and-click interface to construct complete applications from software components. LBNL has multiple research efforts aimed at visual programming environments for distributed and parallel computing, including both visualization and computational science.

Application to Multiple Domains. NVAC concepts need to readily accommodate areas of application other than intelligence and/or terrorist related threats. In particular, visualization analytics should be (and should be shown to be) equally useful for application to energy infrastructure (electrical and natural gas) analysis and recovery planning for scenarios involving natural disasters (hurricane, flood, etc.) – before the fact, during the incident (real-time), and after the fact. INEEL can contribute change detection software, information visualization by query (IVBQ), internet data minder (IDM) and fractal based image analysis.

Network security and analysis. The Bro project[27] from LBNL implements what can be thought of as a "reactive firewall" that detects threats in real time on network borders and dynamically blocks connections deemed to be hazardous. In contrast to a traditional firewall, which uses an exclusionary policy for projection, Bro monitors network connections in real time and takes action on connections deemed to be a threat. Although Bro performs its own analysis in real time along with active control of network connections, Bro as well as with other network infrastructure elements (routers, switches, etc.), generate streams of data that can be subsequently used as input to analysis tools. A number of institutions within DOE (LBNL, LANL, LLNL, INEEL) have cybersecurity programs aimed at analysis and visualization of network data.

---

[26] OGSA website: http://www.globus.org/ogsa/
[27] Bro website: http://www.bro-ids.org/

Parallel and statistical computing. The Parallel-R project from ORNL provides parallel statistical and data analysis capabilities, along with the ability to generate graphical output. It implements an interpreted-language interface, and is extensible through the addition of developer-written modules.

5.5 Research and Development

The following is a list the panel felt were relevant and appropriate research topics. Again, the list is simply a sample, and not intended to be exhaustive. In some cases, DOE laboratories already have ongoing research programs that would be relevant to NVAC needs.

- Devise/define an energy infrastructure 'threat' scenario and associated data sets for a real (or hypothetical) situation that can be incorporated, implemented, tested, etc. in the NVAC environment to show value to entities charged with such responsibilities (e.g. DOE Office of Energy Assurance).
- Various test beds at the INEEL (e.g., Cybersecurity, Supervisory Control and Data Acquisition (SCADA), Process Control, Critical Infrastructure Protection) may be appropriate for generating data from real (but controlled and well-defined) scenarios.
- How to get useful results to the people that really need it –a communications and computational issue.
- Determine how to approach uncertainty propagation across various physical and temporal scales in physics based calculations as well as across the visual analytics.
- Build on existing research to use GPUs for hardware-accelerated visualization and analysis on analysts' desktops. (LANL, ORNL, SNL, LLNL)
- Research streaming/incremental algorithms and architectures for NVAC data structures to provide scalability and high-performance from desktops to parallel visualization clusters. (LANL)
- Research effective methods for the visualization and display of ultra large-scale graphs. (LANL)
- Research the use of volume rendering for effective information visualization presentation. (LANL, LBNL, SNL)
- Explore new visual representations of uncertainty, confidence and other statistical properties for use in analysis. (LANL, LBNL)
- Study the application of immersion/multi-modal perception as it relates to NVAC tasks. (LANL, LBNL, INEEL)

6 Panelists

Jim Ahrens, LANL (ahrens@lanl.gov)
Wes Bethel, LBNL (ewbethel@lbl.gov)
Norman Franke, LLNL (franke1@llnl.gov)
Don Jones, PNNL (dr.jones@pnl.gov)

Jim Kohl, ORNL (kohlja@ornl.gov)
Tom Larson, INEEL (tkl@inel.gov)
Steve Smith, LANL (sas@lanl.gov)
Andy Wilson, SNL (atwilso@sandia.gov)
Chance Younkin, PNNL (chance.younkin@pnl.gov)

Research supported by the Mathematics, Information and Computational Sciences Office, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.