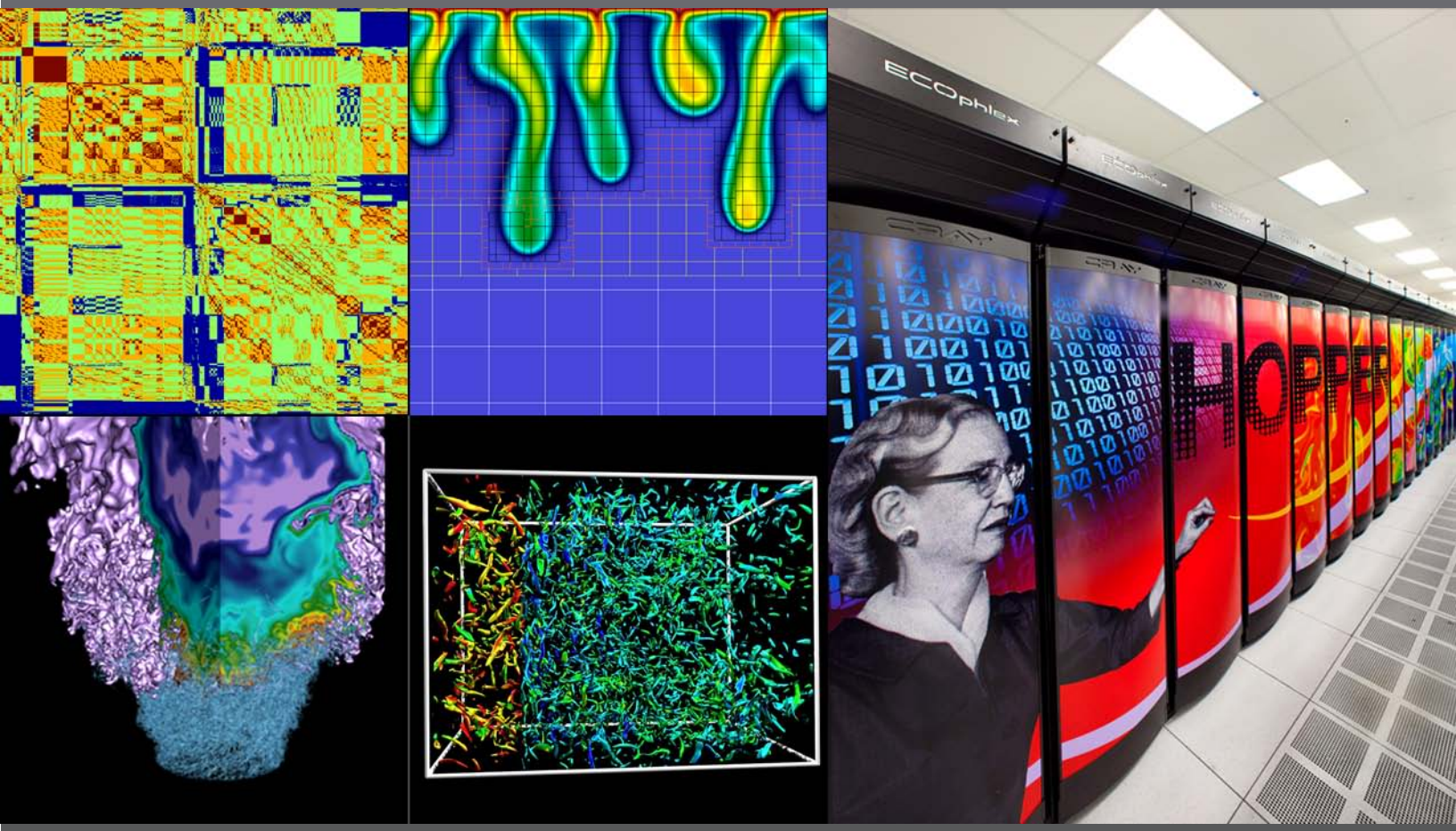
LARGE SCALE COMPUTING AND STORAGE REQUIREMENTS



Advanced Scientific Computing Research

Report of the NERSC / ASCR
Requirements Workshop
January 5 and 6, 2011

DISCLAIMER

Ernest Orlando Lawrence Berkeley National Laboratory

University of California

Berkeley, California 94720 U.S.A.

NERSC is funded by the United States Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) program. Yukiko Sekine is the NERSC Program Manager and Karen Pao serves as the ASCR allocation manager for NERSC.

NERSC is located at the Lawrence Berkeley National Laboratory, which is operated by the University of California for the US Department of Energy under contract DE-AC02-05CH11231.

This work was supported by the Director of the Office of Advanced Scientific Computing Research, Facilities Division.

This is LBNL report LBNL-5249E published January 2012.

Large Scale Computing and Storage Requirements for Advanced Scientific Computing Research

Workshop Report
Conducted January 5 – 6, 2011
Oakland, California

DOE Office of Science

Office of Advanced Scientific Computing Research (ASCR)

National Energy Research Scientific Computing Center (NERSC)

Editors

Richard A. Gerber, NERSC

Harvey J. Wasserman, NERSC

Table of Contents

1 Executive Summary 6

2 About the Office of Advanced Scientific Computing Research (ASCR) 7

3 About NERSC 9

4 Workshop Background and Structure 11

5 Workshop Demographics 12

5.1 Participants 12

5.2 NERSC Projects Represented by Case Studies 13

6 Findings 14

6.1 Summary of Requirements 14

6.1.1 ASCR projects will need more than 1 billion hours of computing time at NERSC in 2014 to meet their research goals and help enable world-class scientific discovery at Office of Science HPC facilities. 14

6.1.2 Applications will need to be able to read, write, and store 100s of terabytes of data for each simulation run. Many petabytes of long-term storage will be required to store and share data with the scientific community. 14

6.1.3 Access to appropriate resources and support for workflows involving many small and medium-sized runs is required. 14

6.1.4 ASCR projects need access to, and robust support for, a rich set of software applications, libraries, and tools. 15

6.2 Other Significant Observations 15

6.3 Computing Requirements 16

7 NERSC Initiatives and Plans 17

7.1 Compute Resources 17

7.2 Data 18

7.3 Software 19

8 HPC Applications and Uncertainty Quantification 20

8.1 HPC Applications and Uncertainty Quantification Overview 20

8.2 Applications and Uncertainty Quantification Case Studies 22

8.2.1 Compressible Turbulence and its Interaction with Shock Waves and Material Interfaces 22

8.2.2 Simulation and Analysis of Reacting Flows 25

8.2.3 The SciDAC Applied Differential Equations Center (APDEC) 29

8.2.4 Non-Intrusive Uncertainty Quantification for Multi-physics Models 34

9 Visual Analytics and Data Management 37

9.1 Visual Analytics and Data Management Overview 37

9.2 Visual Analytics and Data Management Case Studies 39

9.2.1 Data Analytics and Visualization 39

9.2.2 Visualization and Analysis of Volume and Particle Data from Turbulent Combustion Simulations 44

10 Math Software 47

10.1 Math Software Overview 47
10.2 Math Software Case Studies 47
10.2.1 High Performance Sparse Matrix Algorithms 47
10.2.2 DOE Advanced CompuTational Software Collection 49
11 Computer Science and Performance Evaluation 51
11.1 Computer Science and Performance Evaluation Overview 51
11.2 Computer Science and Performance Evaluation Case Studies 51
11.2.1 Advanced HPC Programming Technologies Center 51
11.2.2 UPC, CAF, and Titanium 54
Appendix A. Attendee Biographies 57
Appendix B. Workshop Agenda 61
Appendix C. Abbreviations and Acronyms 62
Appendix D. About the Cover 63

1 Executive Summary

The National Energy Research Scientific Computing Center (NERSC) is the primary computing center for the DOE Office of Science, serving approximately 4,000 users and hosting about 550 projects that involve nearly 700 codes for a wide variety of scientific disciplines. In addition to large-scale computing resources NERSC provides critical staff support and expertise to help scientists make the most efficient use of these resources to advance the scientific mission of the Office of Science.

In January 2011, NERSC and DOE's Office of Advanced Scientific Computing Research (ASCR) held a workshop to characterize HPC requirements for ASCR research over the next three to five years. The effort is part of NERSC's continuing involvement in anticipating future user needs and deploying necessary resources to meet these demands.

The workshop revealed several key points, in addition to achieving its goal of collecting and characterizing computing requirements. Chief among them:

1. ASCR projects will need more than one billion hours of computing time at NERSC in 2014 to meet their research goals and help enable world-class scientific discovery at Office of Science HPC facilities. Approximately one-half of these hours are self-reported requirements for visual analytics.
2. Applications will need to be able to read, write, and store 100s of terabytes of data for each simulation run. Many petabytes of long-term storage will be required to store and share data with the scientific community.
3. Access to appropriate resources and support for workflows involving many small and medium-sized runs is required.
4. ASCR projects need access to, and robust support for, a rich set of software applications, libraries, and tools.

This report expands upon these key points and adds others. The results are based upon representative samples, called "case studies," of the needs of science teams within ASCR. The case studies were prepared by ASCR workshop participants and contain a summary of science goals, methods of solution, current and future computing requirements, and special software and support needs. Participants were also asked to describe their strategy for computing in the highly parallel, "multi-core" environment that is expected to dominate HPC architectures over the next few years.

The report also includes a section with NERSC responses to the workshop findings. NERSC has many initiatives already underway that address key workshop findings and all of the action items are aligned with NERSC strategic plans.

2 About the Office of Advanced Scientific Computing Research (ASCR)

The Advanced Scientific Computing Research (ASCR) program discovers, develops, and deploys the computational and networking capabilities that enable researchers to analyze, model, simulate, and predict complex phenomena important to the Department of Energy. Advanced mathematics and computing provide the foundation for models and simulations, which permit scientists to gain insights into problems ranging from bioenergy and climate change to Alzheimer's disease. ASCR and its predecessor programs have led these advances for the past thirty years by supporting the best applied math and computer science research, delivering world class scientific simulation facilities, and working with discipline scientists to deliver exceptional science.

ASCR's basic research and computing facilities are world class. The Research Division supports research and development in Applied Mathematics, Computer Science, and Next Generation Networks. The Research Division disseminates and further expands ASCR's computational expertise and intellectual resources through its Computational Partnerships with science organizations in the Office of Science. These partnerships are realized through SciDAC Institutes (see the inset below about the SciDAC program), SciDAC Scientific Computations Applications Partnerships, and Exascale Co-Design.

The Facilities Division is responsible for three supercomputing facilities – facilities that house some of the world's fastest supercomputers -- at the Oak Ridge Leadership Computing Facility (OLCF), the Argonne Leadership Computing Facility (ALCF), and the National Energy Research Scientific Computing Center (NERSC), as well as the Energy Sciences Network (ESnet) that facilitates scientific collaborations and the sharing of scientific data. ASCR is guided by science needs and requirements of applications that are critical to the DOE and the nation. Today, modeling and simulation are integral parts of the "scientific method." A good simulation can inform experiment design to assure the return of high-quality experimental data. A high-fidelity simulation is indispensable for the analysis of physical phenomena. The demand for scientific rigor and defensibility in modeling and simulation requires verification, validation, and uncertainty quantification (V&V and UQ). With the unprecedented data available today and becoming even more available in the future (from both observations and simulations), data analytics and data management are rapidly gaining importance as interdisciplinary research and development areas. The demand for computing resources, in terms of processor hours, data transmission and storage, application software, workflow, analysis tools, HPC support, etc., will only increase.

At the same time, high-performance computers are undergoing architectural changes. It is generally agreed that power density already limits the frequency at which semiconductor chips can operate, and future simulation speed-ups will come from exploiting increased fine-grained parallelism on energy-efficient many- and multi-core processors. These constraints of physics and engineering are forcing a paradigm shift in the way scientific simulation and analysis codes are written and executed. Instead of optimizing the total number of calculations, programmers will need to put a premium on maximizing data locality and reducing data movement. Many computational scientists have not experienced such a paradigm shift. This may spur changes in basic understanding of algorithms, operating systems, programming models, performance analysis and tuning– in other words, to prepare for exascale computing, much of the basic research supported by ASCR may undergo significant changes in directions and focus as well!

Today ASCR is at a critical juncture. Demand for high-performance computing is increasing rapidly at the same time high-performance computing paradigms are changing radically. It is in this climate that this requirement workshop attempts to cover areas of ASCR research and development that may become more prominent in the future and help understand computing needs in a rapidly changing landscape.

The SciDAC program was initiated in 2001 as a partnership involving all of the Office of Science (SC) program offices to dramatically accelerate progress in scientific computing that delivers breakthrough scientific results through partnerships comprised of applied mathematicians, computer scientists, and scientists from other disciplines. The SciDAC-2 projects, re-competed in 2006, have now ended. In 2011, ASCR-funded SciDAC-3 Institutes were re-competed; three projects, focusing on applied mathematics, computer science, and uncertainty quantification, were selected. Competitions for targeted Scientific Computation Application Partnerships are ongoing and selections are expected later in FY12.

Through partnerships with ASCR-funded mathematicians and computer scientists, SciDAC applications pursued computational solutions to challenging problems in climate science, fusion research, high energy physics, nuclear physics, astrophysics, material science, chemistry, particle accelerators, biology and the reactive subsurface flow of contaminants through groundwater.

Today the SciDAC program is recognized as the leader in accelerating the use of high-performance computing to advance the state of knowledge in science applications.

3 About NERSC

The National Energy Research Scientific Computing (NERSC) Center, which is supported by the U.S. Department of Energy's Office of Advanced Scientific Computing Research (ASCR), serves more than 4,000 scientists working on about 550 projects of national importance. Operated by Lawrence Berkeley National Laboratory (LBNL), NERSC is the primary high-performance computing facility for scientists in all of the research programs supported by the Department of Energy's Office of Science. These scientists, working remotely from DOE national laboratories; universities; other federal agencies; and industry, use NERSC resources and services to further the research mission of the Office of Science (SC). While focused on research that supports DOE's missions and scientific goals, computational science conducted at NERSC spans a range of scientific disciplines, including physics, materials science, energy research, climate change, and the life sciences. This large and diverse user community runs hundreds of different application codes. Results obtained using NERSC facilities are citied in about 1,500 peer reviewed scientific papers per year. NERSC activities and scientific results are also described in the center's annual reports, newsletter articles, technical reports, and extensive online documentation. In addition to providing computational support for projects funded by the Office of Science program offices (ASCR, BER, BES, FES, HEP and NP), NERSC directly supports the Scientific Discovery through Advanced Computing (SciDAC[1]) and ASCR Leadership Computing Challenge[2] Programs, as well as several international collaborations in which DOE is engaged. In short, NERSC supports the computational needs of the entire spectrum of DOE open science research.

The DOE Office of Science supports three major High Performance Computing Centers: NERSC and the Leadership Computing Facilities at Oak Ridge and Argonne National Laboratories. NERSC has the unique role of being solely responsible for providing HPC resources to all open scientific research areas sponsored by the Office of Science. The Leadership Computing Facilities support a more limited number of select projects, whose research areas may not span all Office of Science objectives and are not restricted to mission-relevant investigations.

This report illustrates NERSC alignment with, and responsiveness to, DOE program office needs, in this case, the needs of the Office of Advanced Scientific Computing Research. The large number of projects supported by NERSC, the diversity of application codes, and its role as an incubator for scalable application codes present unique challenges to the center. As demonstrated by the overall scientific productivity by

[1] http://www.scidac.gov
[2] http://www.sc.doe.gov/ascr/incite/AllocationProcess.pdf

NERSC users, however, the combination of effectively managed resources and excellent user support services, the NERSC Center continues its 35-year history as a world leader in advancing computational science across a wide range of disciplines.

For more information about NERSC visit the web site at http://www.nersc.gov.

4 Workshop Background and Structure

In support of its mission and to maintain its reputation as one of the most productive scientific computing facilities in the world, NERSC regularly collects user requirements from a variety of sources. Methods include scrutiny of the NERSC Energy Research Computing Allocations Process (ERCAP) allocation requests to DOE; workload analyses; and discussions with DOE program managers and scientist customers who use the facility.

In January 2011, NERSC and the DOE Office of Advanced Scientific Computing Research (ASCR, which manages NERSC), held a workshop to gather HPC requirements for current and future science programs funded by ASCR. This report is the result.

This document presents a number of consensus findings. The findings are based upon a selection of case studies that serve as representative samples of NERSC research supported by ASCR. The case studies were chosen by the DOE Program Office Manager and NERSC personnel to provide broad coverage in both established and incipient ASCR research areas. Since ASCR supports many research endeavors in these fields the case studies presented here do not represent the entirety of ASCR research.

Each case study contains a description of scientific goals for today and for the future, a brief description of computational methods used, and a description of current and expected future computing needs. Since supercomputer architectures are trending toward systems with chip multiprocessors containing hundreds or thousands of cores per socket and perhaps millions of cores per system, participants were asked to describe their strategy for computing in such a highly parallel, "multi-core" environment.

Requirements presented in this document will serve as input to the NERSC planning process for systems and services, and will help ensure that NERSC continues to provide world-class resources for scientific discovery to scientists and their collaborators in support of the DOE Office of Science, Office of Advanced Scientific Computing Research.

NERSC has been conducting requirements workshops for each of the six DOE Office of Sciences offices that allocate time at NERSC (ASCR, BER, BES, FES, HEP, and NP). The process began in May 2009 (with BER) and concluded in May 2011 (with NP). The target for science goals and computing requirements has been approximately 2013 or 2014 for each workshop.

Specific findings from the workshop follow**.**

5 Workshop Demographics

5.1 Participants

Name	Institution	Area of Interest	NERSC Repo(s)
Yukiko Sekine	DOE ASCR	NERSC Program Manager	
Karen Pao	DOE ASCR	Applied Math Program Mgr	
Alok Choudhary	Northwestern Univ.	Parallel I/O	m844
Erich Strohmaier	LBNL	Computer Science & Performance Evaluation	m1270
Osni Marques	LBNL	Math Software	m340
Esmond Ng	LBNL	Math Software	mp127
Arie Shoshani	LBNL	Data and Analytics	sdmstor
Wes Bethel	LBNL	Data and Analytics	m636
Kwan-Liu Ma	UC Davis	Data and Analytics	
Nagiza Samatova	NC State Univ.	Data and Analytics	
Rob Ross	ANL	Data and Analytics	
Charles Tong	LLNL	Applications - Uncertainty Quantification	
Johan Larsson	Stanford University	Applications	m837
Sanjiva Lele	Stanford University	Applications	m837
John Bell	LBNL	Applications	mp111, m1055
Richard Gerber	NERSC	Workshop Facilitator	
Harvey Wasserman	NERSC	Workshop Facilitator	
Kathy Yelick	NERSC	NERSC Director	

5.2 NERSC Projects Represented by Case Studies

The NERSC projects represented by the workshop case studies are listed in the table below, along with the number of NERSC hours used by those projects in 2010. The workshop attendees represented a large fraction of the ASCR research performed at NERSC, with 85% of the hours used by NERSC ASCR projects were represented by the Principle Investigator or a senior researcher. Two additional projects, m888, "NERSC / Berkeley Lab Advanced Technology Group Center of Excellence," and m1055, "APDEC Applied Differential Equations Center," are also included in the report.

NERSC Project ID (Repo)	NERSC Project Title	Principal Investigator	Workshop Speaker	Hours Used at NERSC in 2010
m888	Advanced HPC Programming Technologies Center	John Shalf	-	5.6 M
m837	High-fidelity simulations of supersonic turbulent mixing and combustion	Sanjiva Lele	Johan Larsson Sanjiva Lele	4.6 M
mp111	Simulation and Analysis of Reacting Flows	John Bell	John Bell	4.4 M
m636	SciDAC2 Visualization and Analytics Center for Enabling Technologies	Wes Bethel	Wes Bethel	1.4 M
m1055	APDEC: Applied Differential Equations Center (SciDAC)	Phillip Colella	-	1.0 M
mp127	High Performance Sparse Matrix Algorithms	Esmond Ng	Esmond Ng	0.4 M
mp215	UPC, CAF and Titanium	Katherine Yelick	Erich Strohmaier	0.4 M
m340	Installation, Testing and Evaluation of ACTS Tools	Osni Marques	Osni Marques	10 K
Total Represented by Case Studies				**17.8 M**
All ASCR at NERSC in 2010				**21 M**
Percent of NERSC ASCR Represented by Case Studies				**85%**

6 Findings

6.1 Summary of Requirements

The following is a summary of consensus requirements derived from the case studies.

6.1.1 ASCR projects will need more than 1 billion hours of computing time at NERSC in 2014 to meet their research goals and help enable world-class scientific discovery at Office of Science HPC facilities.

a) This is 52 times more than ASCR projects used at NERSC in 2010.
b) About one-half of the hours PIs report they need are for a data visualization and analytics project.
c) Increased time is needed for development, testing, and implementation of petascale applications at scale.
d) The need for in-situ visualization and data analysis will require at least a 15% increase in hours used by large-scale applications.
e) The growing demand for code verification and validation, along with uncertainty quantification, will require at least an additional 10% increase in available computing hours.
f) A need for 1 billion hours in 2014 is consistent with historical trends in the growth of ASCR usage at NERSC since 2002.

6.1.2 Applications will need to be able to read, write, and store 100s of terabytes of data for each simulation run. Many petabytes of long-term storage will be required to store and share data with the scientific community.

a) Data analysis and visualization files will be large as world-class simulations continue to grow.
b) Extremely large checkpoint files, written often, will be required to guard against job failures.
c) Projects need long-term storage of massive datasets for distribution to the scientific community.

6.1.3 Access to appropriate resources and support for workflows involving many small and medium-sized runs is required.

a) Scientific codes and results derived from their runs will be subject to greatly increasing demands for verification, validation, and uncertainty quantification. This will require unprecedented number of runs at less than full scale.
b) Post-processing data analysis and visualization are required and will not be replaced by in-situ analysis. Some analytic applications are I/O and memory intensive (8 GB/core).

6.1.4 ASCR projects need access to, and robust support for, a rich set of software applications, libraries, and tools.

a) An optimized version of the HDF5 I/O library is crucial.
b) The VisIt visualization application is needed by many projects. In-situ analysis and post-processing of massive data sets require that VisIt can run at scale and on the same platform as the scientific applications.
c) Optimization tools and libraries are required to enable scientific applications to scale to petascale and exascale. These include PAPI, Tau, IPM, CrayPat, HPCToolkit.
d) Tools for efficiently handling petabytes of data are needed.

6.2 Other Significant Observations

The following are topics NERSC observed at the workshop and in the case studies. All of these have been heard in workshops held with other Office of Science program offices.

- HPC system architectures are expected to continue trending towards using huge numbers of low-power, simple cores, with increasing numbers of core per CPU socket. This will necessitate a shift in programming methods to use these machines, but the future of programming is vastly uncertain. The day of pure MPI codes is likely over, even for mid-range computing. The overhead of MPI is too high per task and traditional methods of programming with MPI use too much memory per core. The most common first approach being used by projects is to augment coarse-grained MPI parallelism with OpenMP threading, which enables fine-grained parallelism and encourages memory reuse among threads. Other projects are investigating the so-called Partitioned Global Address Space (PGAS) languages and programming GPUs using CUDA, CUDA Fortran, /OpenCL, or directive-based solutions. There is some skepticism that OpenMP will be an appropriate programming model for expressing and achieving performance.

- Rewriting codes based on a new programming model is expensive and scientists don't want to invest in unproven and uncertain technologies. In a university scientific research setting there is a reluctance to spend student and post-doc time reprogramming codes, which delays publication and graduation rates, and may result in programs that become quickly obsolete if a competing programming model wins out.

- Most scientific applications can function well with 1-2 GB of memory per core, but analysis codes benefit greatly, and may require, up to 8 GB/core. Because many large NERSC projects compute on a variety of machines, then tend have version that can run in low-memory environments.

- Projects would like development and hosting of code development, collaboratory, and data-sharing interfaces and portals. Examples include web portal hosting, and hosting code development repositories (e.g. subversion, Git).

- Access to prototype architectures is important to ASCR projects for developing new languages, algorithms, and analysis tools.

6.3 Computing Requirements

The following table lists the computational hours required by research projects represented by case studies in this report. One project did not have an allocation of resources at NERSC in 2010. "Total Scaled Requirement" at the end of the table represents the hours needed by all 2010 ASCR NERSC projects if increased by the same factor as that needed by the projects represented by case studies in this report. About one-half of the hours PIs report they need are for the project marked with an asterisk (*).

NERSC Project Title	Principal Investigator	Hours Needed in 2014	Increase Over 2010 NERSC Use
Compressible Turbulence and its Interaction with Shock Waves and Material Interfaces	Lele	100 M	22
Simulation and Analysis of Reacting Flows	Bell	50 M	11
Data Analytics and Visualization*	Bethel	500 M	360
High Performance Sparse Matrix Algorithms	Ng	2 M	5
DOE Advanced CompuTational Software Collection	Marquez	250 K	250
Applications and Uncertainty Quantification	Various	No direct needs at NERSC	Additional 10% for science runs
APDEC: Applied Differential Equations Center (SciDAC)	Collela	60 M	60
Advanced HPC Programming Technologies Center	Shalf	36 M	5.8
UPC, CAF, and Titanium	Yelick	10 M	26
Visualization and Analysis of Volume and Particle Data from Turbulent Combustion Simulations	Ma		Additional 15% for science runs for *in-situ* vis
Total Represented by Case Studies		758 M	42
Percent of ASCR Workload Represented		85%	
Additional 25% for UQ/V&V and In-Situ Vis		x 1.25	
All ASCR at NERSC Total Scaled Requirement		**1,115 M**	**42**

7 NERSC Initiatives and Plans

NERSC has initiatives already underway and long-term strategic plans that address some requirements presented in this report. A brief summary of these initiatives and plans is presented in this section.

7.1 Compute Resources

The NERSC Hopper system, a Cray XE6 with 1.3 PF/s of peak performance and 120 TF/s performance on a set of representative applications, was installed in the fall of 2010 and went into production on May 1, 2011. Hopper represented a 4-fold increase in aggregate application performance over the quad-core Franklin system that went into production in mid-2009. Total allocations in 2011 and 2012 are expected to be about 1.1B hours (a factor of 3.5 over 2010). NERSC has started the procurement process for a NERSC-7 system, which we anticipate will be available for production computing by 2014 at the latest.

Current technology trends, along with the estimated NERSC funding profile, suggest that NERSC will not be able to meet the demand stated in this report by 2014. The figure below, showing the historical growth of ASCR and overall usage at NERSC, indicates that the need for computational hours to support ASCR in 2014 slightly exceeds that expected by the historical trend (lower black line) whereas the total number of hours expected at NERSC, based on current plans and funding profiles, is not growing as fast as that trend.

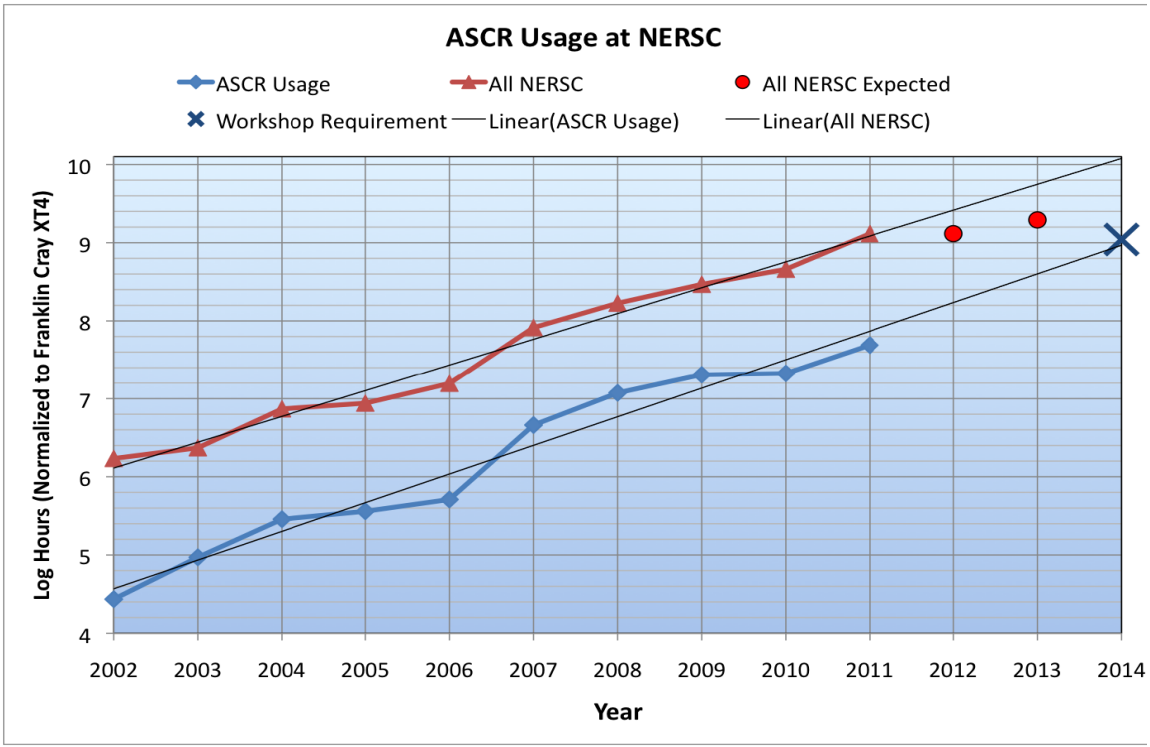


Figure 1: Total usage and ASCR usage at NERSC per year. The red circles represent expected number of hours at NERSC in future years, based on expected funding. The large blue X denotes the hours needed by ASCR at NERSC, based on this report.

There is currently great interest in using GPUs to accelerate computations. NERSC has provided users with the Dirac testbed for exploring this architectural path. There has been considerable activity in studying the applicability of GPU-based hardware solutions with impressive performance gains in some CPU-intensive kernels, our NERSC users have told us that they are not yet prepared to make a substantial investment in a technology that may not be appropriate for their workload. NERSC will be closely monitoring these and other low power processor and memory technologies, communicating with vendors to help them understand the needs of the full workload, and working with the ASCR community to better understand ways of adapting their algorithms and software to future systems.

7.2 Data

NERSC plans to continue a constant investment in storage each year; at the planned budget levels this would result in a four-fold increase in disk capacity in the next four years. Based on the findings in this report NERSC is largely on track to meet ASCR data requirements, although it should be noted that some ASCR projects are projecting the need for 100-300 TB online quotas – which represents a 5-to-15-fold increase over

today's largest quotas versus our expected 4-fold projected growth. NERSC is investigating the use of group quotas to satisfy such needs.

NERSC is also investing heavily in improving both capacity and bandwidth for the HPSS archival storage system. In 2011, NERSC added increased bandwidth to achieve 10 percent of the fastest file system's aggregate bandwidth. NERSC is also adding a tape library to increase its archival storage capacity. The NERSC HPSS system is designed to handle 50% growth per year in amount of I/O and total data stored. The system handled over 4 PB of I/O in 2010 and grew by 2.2 PB. Additional capacity is expected to grow by a factor of two each year for the next three to five years. This is in line with conventional bandwidth guidelines at other centers. NERSC is also working to significantly improve data movement between HPSS and NGF.

NERSC is also working closely with ESnet to implement the Advanced Networking Initiative-based 100Gb networks that will bring about significant improvement – 10X – in data movement capability. NERSC also supports GlobusOnline capability and a dedicated set of data transfer servers for rapid file transfers. For the foreseeable future (~five years) these capabilities are likely to represent the primary methods for data transfer between sites.

7.3 Software

NERSC recognizes the importance of providing scientific software libraries and is committed to encouraging vendors to install and support these popular libraries and ensure that they're optimized for the production platforms. Cray currently provides optimized versions of PETSc, Global Arrays, Trilinos, HDF5, netCDF, FFTW, and a large suite of mathematical libraries. The DOE ACTS tools group also provides optimized versions of important software and works with NERSC to provide user support when needed.

The HDF5 library is one of the most commonly used I/O libraries at NERSC and DOE. For this reason, NERSC partnered with the nonprofit Hierarchical Data Format (HDF) Group to optimize the performance of the HDF5 library on modern HPC platforms. This effort is continuing through UCSD. NERSC is also actively engaged with the HPC community in improving I/O performance. Efforts have been focused on MPI-IO, file caching/prefetching/aggregation, and other areas.

8 HPC Applications and Uncertainty Quantification

8.1 HPC Applications and Uncertainty Quantification Overview

Ten years ago, the U.S. Department of Energy Office of Science launched an innovative software development program with a straightforward name — Scientific Discovery through Advanced Computing, or SciDAC. The goal was to develop scientific applications to effectively take advantage of the terascale supercomputers then becoming widely available. Today, both the program and its goal are major focuses of the Office of Advanced Scientific Research as supercomputers move to the petascale and exascale.

Many of the biggest users of ASCR allocations at NERSC are working on applications based on solving partial differential equations (PDEs). PDEs are used heavily in turbulence, combustion, and astrophysics modeling. Typically, these applications deploy mature numerical methods whose properties are well understood either mathematically or empirically, with application-specific modifications. The major computational challenge today is scalability, both for running at NERSC and competing for INCITE allocation time on either the OLCF or the ALCF.

Exascale computing brings new challenges. Traditionally, PDE-based applications have expected a 10-fold increase in resolution for each 1,000-ford increase in compute capability, but this is not expected to happen going exascale: the processors will not be 10 times faster and there will not be 1,000 times more memory available. Instead, the increase in computational power will be delivered by massive parallelism and data movement will be the limiting factor. Scalability, therefore, becomes an even more pressing concern for these applications. However, exascale computing also presents an opportunity. Application scientists may incorporate “more physics” instead of increasing resolution to improve the fidelity of the simulation. Exascale computing may also spur the development and deployment of uncertainty quantification (UQ) techniques to establish confidence levels in computed results and deliver predictive science.

For PDE-based applications, memory management will be key in going exascale. According to the report of the workshop on Crosscutting. Technologies for Computing at the Exascale (http://www.exascale.org/iesp/IESP:Documents), mathematicians may need to consider the following in designing algorithms for exascale applications:

- Recast applied math algorithms/PDE discretization to reflect shift from FLOP-centric to memory-constrained hardware;
- New algorithms with more compute, less communication;
- Applications must care more about fault tolerance and resilience; and
- Checkpoint-Restart may not work with current storage & I/O systems.

In the decade of stockpile stewardship ── that is, no new, full-up, integral nuclear experiments ── using simulation and modeling become the main tools to ensure the safety and reliability of existing nuclear stockpile. However, since most experimental data are "integral" in nature, they do not have a simple relationship to the models used in the simulations, and in general do not directly validate any individual physics model in the multi-physics context. Thus the need to understand the uncertainties – being able to quantify the confidence in the modeling and simulation efforts that lead to an annual certification -- in nuclear weapons application modeling and simulations became more obvious as experiments become harder and more expensive to conduct.

An emphasis on Uncertainty Quantification (UQ) is a relatively new development for the Office of Science. The need for UQ for the climate modeling community has recently become clear: like the weapons community, no full-up integral experiment can be conducted, and like the nuclear weapons community, many uncertainties exist in individual "physics" models and the data can only "validate" integrated systems. And, like the weapons community, there is a need to be able to express the level of confidence in the simulation and modeling effort.

UQ has not historically been a major concern for scientific discovery, but as the cost of computing resources becomes higher, UQ will become more important as a mechanism to validate the fidelity and societal impact of simulations.

Uncertainty quantification is on the critical path of ASCR's March to Exascale. In 2010, ASCR made a modest investment in mathematical methods of uncertainty quantification. These are ongoing research projects with limited scopes. Even though today there are few UQ activities on ASCR facilities, it is anticipated that these activities will increase over time. For the UQ Case Study we are going to look at an example from a programmatic application at a weapons lab as a reference point on possible requirements for science applications.

Three case studies in applications and uncertainty quantification follow.

8.2 Applications and Uncertainty Quantification Case Studies

8.2.1 Compressible Turbulence and its Interaction with Shock Waves and Material Interfaces

Principal Investigator: Sanjiva Lele (Stanford University)
Contributors: Johan Larsson (Stanford University)
NERSC Repository m837

8.2.1.1 Summary and Scientific Objectives

The focus of this work (sponsored under the DOE SciDAC-2 program) is on compressible turbulence and the fundamental physics of shock/turbulence interactions and shock/material interface interactions. The goal is to elucidate the underlying physics of these processes, which will lead to better engineering models. The work is applicable to a wide range of problems, including high-speed flight and propulsion, astrophysics, inertial confinement fusion, and medical procedures involving shocks (e.g., removal of kidney stones).

The research is mainly focused on canonical problems, including the interaction between isotropic turbulence and a normal shock wave. This problem is difficult to study experimentally, which implies that high-fidelity simulations have a major role to play.

8.2.1.2 Methods of Solution

We solve the compressible Navier-Stokes equations for a perfect gas (or a mixture of perfect gases). The numerical method solves the weak form of the equations, i.e., it captures discontinuities (shocks and material interfaces) correctly. The method is a rather novel solution-adaptive algorithm, which applies different numerics to discontinuities and broadband turbulence. The method was first developed and implemented in the *Hybrid* code, which is a research code limited to Cartesian geometries (e.g., canonical problems). In subsequent work, the algorithm was modified for general-purpose, unstructured-grids and implemented in the *Charles* code. Both codes are parallelized using MPI, including I/O. The algorithm is fully explicit in both space and time, and the implementation uses fully non-blocking communication. This leads to very good parallel scaling characteristics; e.g., above 98% weak scaling efficiency when going from 8 to 65,536 cores on the BG/P machine at ALCF.

The memory footprint of these codes is small for most computations: between 60 and 500 doubles per grid point depending on the code (structured or unstructured) and problem (single or multiple gases). The exception to this is for combustion simulations, where the

combustion chemistry is pre-computed and tabulated. The chemistry table typically requires about 1 GB of storage space (kept in memory).

8.2.1.3 HPC Requirements

We run a spectrum of different types of simulations. In 2010, some typical runs were:

- *Big (3 runs)*: 2.5 billion grid points, run on either 65,536 cores on the BG/P machine at ALCF for about 18 hours or on 12,288 cores on the Cray XT-4 at NERSC for about 48 hours. These runs generated about 2 TB of data each.
- *Small (hundreds of runs)*: 2 – 50 million grid points, run on 64 – 512 cores either on a Dell cluster at Stanford or on the Cray XT-4 at NERSC.
- *Post-processing (excluding visualization)*: e.g. sequentially reading in 100 snapshot files, computing various statistics, filtering, etc. This is a completely memory-limited process, with the number of cores chosen to fit in memory (typically 256-512 cores for a few hours). This involves a very interactive and iterative work-flow: e.g., first computing some quantity, when looking at this result realizing that we should compute some other quantity, etc.

In 2010 we used a total of 12M hours on the BG/P at ALCF, 4.6M hours on the Cray XT-4 at NERSC, and very roughly 1M hours on the Dell-cluster at Stanford.

We also note that we store the computed data (snapshots of the solution) for several years, and go back and re-analyze the data (to answer slightly new questions) several times. Therefore long-term storage is very important to us.

By 2014 our research and computational requirements will have changed in the following ways:

- *Fundamental shock/turbulence physics:* To test our recent hypotheses on the physics of the interaction process (that have come out of our current large-scale runs), we need to run a few (two or three) very large simulations. The grids would have 4-8 times more grid points, so 10B-20B grid points. On the current Cray XT-4 architecture, this implies 3M-9M core-hours per run, using very little memory (about 4 TB aggregate memory). We would need to store about 10 TB of data per run.
- *Modeling development and validation:* Predictive models are very important for engineering (e.g., design of high-speed vehicles or ICF technologies) and science applications (e.g., investigations into supernovae explosions). Development and validation of improved predictive models will require many small-to-medium sized runs, in the range of 2M – 200M grid points (higher numbers for more complex/realistic geometries). On current architectures, these runs would require 10-4,000 cores.
- *Reactive flow modeling:* For chemically reacting high-speed turbulent flows (e.g., nuclear reactions in ICF and supernovae, fuel/oxidizer in propulsion devices) we are using tabulated pre-computed chemistry. These tables are read-only, accessed very frequently (every grid point, every time step) and currently require roughly 1

GB of memory. While we currently keep one copy of the table in memory for each core, this is not necessary – it would suffice to keep one copy in memory for each node (on current architectures). Thus the main additional requirement from these types of runs is the need for at least 2 but preferably 4 GB of memory per shared-memory node.

The overall increase in core-hours and disk-usage is easiest to estimate from our biggest runs (item 1 above): these are 5-10 times larger than current runs, both in terms of core-hours, memory-usage, and disk-usage.

8.2.1.4 Computational and Storage Requirements Summary

	Current (2010 NERSC)	2014 Requirement
Computational Hours	17 M	100 M
Parallel Concurrency	66 K (big), 2 K (medium)	500 K (big), 10 K (medium)
Wall Hours per Run	12 – 24	24 – 48
Aggregate Memory	1 TB (big)	4 TB (big)
Memory per Core	16 MB	16 MB
I/O per Run	1 – 2 TB (big)	10 TB (big)
On-Line Storage Needed	10 TB	100 TB
Data Transfer		
Archival Storage	10 TB	200 TB (accrued)

8.2.1.5 Support Services and Software

We currently use the VisIt visualization software, and the associated Silo library to generate files for visualization. In addition, it is useful (but not strictly necessary) to have access to Matlab on a high-memory portion of a cluster.

8.2.1.6 Emerging HPC Architectures and Programming Models

The PSAAP center at Stanford has a computer science component that is developing a domain-specific language (called Liszt) that can be compiled for CPUs, GPUs, or other architectures. Part of the PSAAP-work will be to implement a module of one of our codes in the Liszt language. In separate work, a student has modified a different legacy code at our Center from pure MPI to hybrid MPI/OpenMP. This was done in an effort to save memory, and has worked out really well.

With respect to the two codes used in the research described here, our strategy has been (and still is) to "wait-and-see" which programming model emerges as the best before undertaking major coding efforts.

8.2.2 Simulation and Analysis of Reacting Flows

PI: John B. Bell (Lawrence Berkeley National Laboratory)
NERSC Repository mp111

8.2.2.1 Summary and Scientific Objectives

The objective of this project is to develop new simulation methodology for multiphysics applications. We use an integrated approach in which we consider the mathematical formulation, discretizations and software issues. In particular, we develop mathematical formulations that reflect the relationship between scales in the underlying problem. We then develop discretizations of those models that incorporate the mathematical structure of the underlying processes. Finally we implement these algorithms in the context of an evolving software infrastructure that facilitates implementation of the methodology on HPC architectures. As a part of this process we conduct scientific investigations in the respective application areas, which are our major consumer of computational resources.

Specific areas we are currently targeting are combustion, porous media flow and astrophysics. The focus of our work in combustion is on high-fidelity simulations of flames with detailed chemistry and transport. We use a low Mach number formulation for these simulations that exploit the separation of scales between the flame dynamics and acoustic wave propagation. The simulations also use adaptive mesh refinement to focus computational resolution near the thin reaction zones.

Our work on subsurface flow focuses on the development of adaptive algorithms for multiphase, multicomponent, non-isothermal flows. Our formulation splits the parabolic behavior of pressure form the advection-dominated behavior of the chemical components of the mixture and the enthalpy. The overall structure of the system is determined by the characterization of the phase equilibrium properties of the mixture. Applications on this methodology include carbon sequestration and environmental remediation.

Our work in astrophysics currently focuses on the simulation of Type Ia supernova (SNIa). Our goal is to provide an end-to-end simulation. Our particular focus is on the simulation of the convection processes leading up to ignition. For these simulations, we use a low Mach number formulation for stratified flows. This capability enables us to follow several hours of the evolution of the star leading up to ignition, which would be infeasible with a standard compressible formulation. The conditions of the star at ignition, including ignition location and the levels of convective turbulence play a critical role in the subsequent explosion of the star. We also have developed a compressible code that includes a turbulent flame model for following the evolution of the star after ignition through the explosion. We are currently developing the methodology to map between these two codes. We are also investigating X-ray bursts, convection in massive stars, core collapse supernovae and cosmological simulations with the methodology we are developing.

8.2.2.2 Methods of Solution

Our work focuses on the development of multiphysics simulation codes within CCSE at LBNL. The names and application areas of these codes are:

- LMC: low Mach number combustion
- PMAMR: porous media
- MAESTRO: low Mach number astrophysics
- CASTRO: compressible astrophysics
- NYX: computational cosmology

While each of these codes is used to simulate flows in a different application area, they share a number of common features:

- Built on CCSE's well-established BoxLib framework
- Rely on iterative linear solvers for constant and/or variable coefficient elliptic and parabolic equations based on geometric multigrid
- Implemented on 2-D or 3-D adaptive grid hierarchies (block-structured grid AMR)

We are currently using a hierarchical parallelization strategy based on a combination of MPI + OpenMP. At a coarse-grained level we distributed patches to nodes using MPI. At a fine grained level within the node we use OpenMP directives to parallelize operations on patches. We have demonstrated that this approach is effective on current HPC architectures and should be extensible to many more cores / node.

8.2.2.3 HPC Requirements

The numerical methodologies we are developing are applicable across a wide range of applications. Here we focus on an example drawn from combustion. Experimental measurements of NO_x emissions from lean premixed hydrogen indicate emission levels higher than would be expected. The computational challenge is to model a turbulent premixed hydrogen flame with sufficient chemical fidelity to identify the chemical pathways that lead to nitrogen emissions and why those pathways lead to higher than expected emissions. This requires simulations that are not only highly resolved spatially but also require a high-fidelity chemical model to capture the kinetics associated with the emissions process. As is typical of our applications, the total number of unknowns is the leading order metric in determining HPC requirements. For the example discussed above, the simulation used approximately 50 billion unknowns. The simulation required several weeks of wall clock time on 20K processors and generated around 10 TB of data.

Over the next few years, we anticipate two types of changes to our methodology:

1. The development of improved discretization approaches. Currently the numerics in our production computations are second-order accurate in both space and time. We plan to develop higher-order versions of our methodology. Higher-order methods will allow us to

trade increased floating-point work for less memory and reduced communication. We will also begin to investigate approaches for incorporating uncertainty quantification into our simulations.

2. The development of in situ analysis capabilities. The increase in compute capability relative to I/O capacity suggests that we need to integrate at least some of the data analysis into the simulation directly to reduce I/O volume.

The types of simulations we anticipate performing in 2014 include:

Combustion: Fully investigate the behavior of alternative fuels with detailed chemistry in the high-pressure environments characteristic of realistic combustion systems. Address key modeling issues related to the development of engineering design models for next generation combustion systems.

Porous media flow: First steps toward quantifying the role of subsurface uncertainty on the prediction of contaminant plumes and carbon sequestration strategies.

Astrophysics: Perform high-fidelity simulations of a variety of different supernova phenomena and compare with observations. Perform detailed cosmological simulations and compare to observations to reduce uncertainty in key cosmological parameters.

Achieving any of these goals would require a balanced growth in the available resources. It would also require a significant shift in our programming models. The two main issues would be effective utilization of higher-levels of intra-node concurrency and tools for enabling in situ analysis of results as part of the overall simulation.

Known bottlenecks: None

Special needs: None

8.2.2.4 Computational and Storage Requirements Summary

	Current (2010 NERSC)	2014 Requirement
Computational Hours	4.4 M	50M
Parallel Concurrency in typical production run	2K – 24K cores	25K – 200K cores
Wall Hours per Run	50-1000	300
Aggregate Memory	1K – 24K GB	10K – 100K GB
Memory per Core	0.5-1.5 GB	.5 – 1.5 GB
I/O per Run Needed	2-60 TB	10-40 TB
On-Line Storage Needed	5 TB	20-40 TB
Archival Storage Needed	100 TB	200 TB

8.2.2.5 Support Services and Software

Important software includes F90, C++, MPI, OpenMP, and htar (for creating a tar file archive directly on the HPSS archive storage system) or similar archiving software. There are two principal areas in which needs for software, services, etc. are anticipated. The first area is improved programming models to support hierarchical parallel approaches (see below). In a similar vein, tools are also needed for automatic performance tuning to improve overall node performance.

The other major area of significant needs is in the area of tools to facilitate archiving simulation data and rapid access to archived data for subsequent analysis. The data volume associated with the simulations discussed elsewhere in this document will be unmanageable without improved data-handling tools.

8.2.2.6 Emerging HPC Architectures and Programming Models

Our methodology is based on a block-structured adaptive mesh refinement strategy. In block-structured AMR methods, regions requiring additional refinement are grouped into large grid patches. Each of these patches is at least 16^3, often larger. This type of approach lends itself naturally to a hierarchical approach to parallelism. We are currently using a combination of MPI + OpenMP. At a coarse-grained level we distributed patches to nodes using MPI. At a fine-grained level within the node we use OpenMP directives to parallelize operations on patches. We have demonstrated that this approach is effective on current HPC architectures and should be extensible to many more cores / node.

However, we note that OpenMP is not an ideal programming model for fine-grained parallelism. We believe that with improved programming models this type of MPI + X strategy would be much more effective. Key issues are:

- A lighter-weight thread model that does not include the overheads of OpenMP programming model support to express data layout to avoid performance penalties associated with non-uniform memory access properties intra-node.

8.2.3 The SciDAC Applied Differential Equations Center (APDEC)

PI: Phillip Colella (Lawrence Berkeley National Laboratory)
Contributor: Dan Martin (Lawrence Berkeley National Laboratory)
NERSC Repository m1055

8.2.3.1 Summary and Scientific Objectives

The Applied Partial Differential Equations Center for Technologies (APDEC) is a SciDAC2 center whose goal to enable the agile development of high-performance simulation codes for complex multiphysics and multiscale applications, by providing a flexible toolset for the development of computer simulations (http://www.apdec.org). In SciDAC3, much of this effort is being continued as a part of the FASTMath institute.

The SciDAC3 FASTMath institute will develop and deploy scalable mathematical algorithms and software tools for reliable simulation of complex physical phenomena, to address challenges facing the scientific computing community in the shift to multi-/many-core nodes and million-way parallelism. Target platforms will primarily be NERSC systems, enabling new scientific results that were previously unobtainable with existing algorithms and architectures.

The overarching theme of these projects is to develop high-resolution finite-volume adaptive methods for partial differential equations, and to implement the resulting methodology into production-quality software tools that are broadly applicable to a number of DOE Office of Science research programs. The project supports active algorithm development and performance optimization, as well as production runs for physics investigations in several target application areas.

One area which we believe will become more important in the time frame relevant to this report is the tendency to combine existing code bases into large "super codes" in order to better model complicated physical systems. While the individual constituent models have likely been verified and optimized for performance on subsets of the problem of interest, it remains an open question how best to combine many sub-models in a consistent and efficient manner in order to compute mathematically reasonable solutions in a computationally efficient manner. This has been done in the climate applications by using coupling approaches that maintain conservation and stability at the cost of accuracy. We will need to develop approaches to couple the higher-order schemes and models currently under development. This will require large-scale runs of the coupled systems for testing and verification.

A non-exhaustive list of efforts supported by this work includes:

- Chombo: A C++ class library designed to support development of scalable high-performance applications that use finite-volume methods and block-structured adaptive mesh refinement. We have helped develop a number of applications that use the Chombo library.

- SciDAC-e: A high performance simulation capability for flowand transport in complex micro-scale geometries with complex geochemistry. In 2012, we will simulate reactive transport processes in realistic pore space, representing a grid resolution of 4 microns, which is at the top end of the current resolution of computed tomographic imagery obtained from the Advanced Light Source (ALS) at LBNL.

- Edge Simulation Laboratory (ESL): jointly funded by the DOE Office of Fusion Energy Science and the DOE Office of Advanced Scientific Computing Research Applied Math base programs to create a continuum gyrokinetic simulation capability suitable for the edge region of tokamak fusion reactors. While the project is specifically oriented to fusion applications, the underlying algorithms are applicable to a wide-range of physical simulation applications.

- Climate: Development of a new dynamic core for climate simulations to achieve an unprecedented 1km resolution with vertical convective motion. Requires new co-development and scaling efforts to enable additional physics and resolution.

- Base Math: ASCR-funded project to develop high-quality tools for simulating complex systems that are represented by solutions to partial differential equations, with multiple physical processes, complex geometries, and multiple length and time scales. Algorithm research to develop multi-core/100k+ core algorithms for specific problems arising in DOE applications.

- BELLA: Development of simulations of laser plasma accelerators under development by the DOE LOASIS program at LBNL. Code development will evolve 1D prototypes to include realistic 3D geometries, complex reactions and compressible (and potentially magneto¬) hydrodynamics.

8.2.3.2 Methods of Solution

Code Name: Chombo

Description: Block Structured Adaptive Mesh Refinement class library and several applications written in Chombo for application stakeholders.

Mathematics: Allspeed method for MHD. Projection method for fluid flow problems. A new 4th order mapped multiblock technology for climate modeling and

astrophysics.

Numerical Techniques: block structured AMR, finite volumes. particle-in-cell techniques.

Machines: Franklin, Hopper

Planned Processors: 20,000 with some runs going as large as Hopper is capable of.

Languages: C++, Fortran90, MPI-IO, OpenMP,PosixThreads

Libraries: Chombo, HDF5, LAPACK, PAPI

Performance Limits: Memory speed, interconnect speed

Checkpoint? Y

Code Plans: Tuning and 200k-500k+ core algorithm development and optimization under FASTMath. We will also be introducing novel hybrid programming models in some parts of the project (MPI+OpenMP, MPI+PGAS, ExaHDF5, etc.).

Eigensolvers: fixed structured, composite, adaptive,none

Other Algorithms: sparse le

Code Name: COGENT

Description: Modeling the transition region between the core and edge plasma reaction regions in a tokamak.

Mathematics: Continuum gyrokinetic simulation capability suitable for the edge region of tokamak fusion reactors, on magnetic-field-aligned grids that efficiently represent the high-dimensional phase space.

Numerical Techniques: The goal of the algorithm research is to develop high-order (fourth), conservative discretizations. The geometric constraints necessitate the development of high-order, mapped multi-block algorithms, and the broad range of temporal scales requires advanced time integration techniques. MPI-based parallelism and algebraic multigrid preconditioned solvers for semi-elliptic operators.

Machines: Franklin, Hopper, Carver

Percent of Allocation: 5

Planned Processors: 100-10k

Languages: C++, Fortran90, MPI, MPI-IO, OpenMP

Libraries: Chombo, HDF5, Hypre, PAPI, PetSC, Tau

Performance Limits: interconnect speed

Checkpoint? Y

Code Plans: Adding capabilities for time-dependent input data files (relatively small data, some impact on computation).

Eigensolvers: fixed structured, composite, adaptive

Other Algorithms: sparse le

Code Name: EBAMRINS

Description: Incompressible Navier-Stokes equations solved on locally adaptive block-structured meshes. This uses the novel technique of ”embedded boundaries” to represent complex geometry problems.

Mathematics: TGA Semi-implicit finite-volume Godunov scheme with redistribution methods at embedded boundary regions. RK4 time integrations.

Numerical Techniques: Structured Mesh, Adaptive Mesh, Finite Volume, Godunov Methods, Projection Schemes. Semi-Implicit time stepping. Lots of least-squares calculations on small systems.

Machines: Franklin, Hopper, Carver

Planned Processors: 100,000 cores

Languages: C++, Fortran90, MPI, MPI-IO, OpenMP

Libraries: Chombo, HDF5, LAPACK, EB-Chombo

Performance Limits: Memory Speed

Performance Comments: Memory load balancing is current limitation

Checkpoint? Y

Code Plans: tuning and potential integration with AMG library (such as PETSC/HYPRE) for some production runs

Eigensolvers: fixed structured, composite, adaptive

Other Algorithms: sparse le

8.2.3.3 HPC Requirements

We anticipate a large increase in production NERSC usage from the current baseline usage for a number of projects. The overarching theme of these projects is to develop high-resolution adaptive methods for partial differential equations, and to implement the resulting methodology into production-quality software tools that are broadly applicable to a number of DOE Office of Science research programs. The project supports active algorithm development and performance optimization, as well as production runs for physics investigations in several target application areas.

Major consumers of cycles will be:

FASTMath: SciDAC program to develop and deploy scalable mathematical

algorithms and software tools for reliable simulation of complex physical phenomena, to address challenges facing the scientific computing community in the shift to multi-/many-core nodes and million-way parallelism. Target platforms will primarily be NERSC systems, enabling new scientific results that were previously unobtainable with existing algorithms and architectures. Using Chombo and EB-Chombo, with other libraries. 30 % of usage.

SciDAC-e: high performance simulation capability for flow and transport in complex micro-scale geometries with complex geochemistry. In 2011, preliminary reactive transport simulations using EB-Chombo have been run on 30-60,000 processors on the NERSC Cray XE6. In order to achieve the objective image resolution and realistic time scales, we will need to scale these to 100,000+ processors. 50 % of usage. In addition, several development efforts will be running prototypes and scaling experiments as they proceed toward production capability:

Edge Simulation Laboratory (ESL): COGENT, a Chombo-based code, runs a continuum gyrokinetic simulation for the edge region of tokamak fusion reactors. Algorithm research is to develop fourth-order, conservative discretizations on magnetic-field-aligned grids, and the broad range of temporal scales requires advanced time integration techniques. 5 % of usage.

Climate: Chombo-based development of a new dynamic core for climate simulations to achieve an unprecedented 1km resolution with vertical convective motion. Requires new co-development and scaling efforts to enable additional physics on an adaptive, mapped multi-block grid. 5 % of allocation.

Base Math: ASCR-funded project to develop high-quality tools for simulating complex systems that are represented by solutions to partial differential equations, with multiple physical processes, complex geometries, and multiple length and time scales. Algorithm research to develop multi-core/100k+ core algorithms for specific problems arising in DOE applications. Development will be on top of Chombo and EB-Chombo. 5 % of usage.

BELLA: Development of simulations of laser plasma accelerators, being developed by the DoE LOASIS program at LBNL. Using EB-Chombo, code development will evolve 1D prototypes to include realistic 3D geometries, complex reactions and compressible (and potentially magneto-) hydrodynamics. 5% of usage.

8.2.3.4 Computational and Storage Requirements Summary

	Current (NERSC 2010)	2014 Requirement
Computational Hours	1 M	60 M
Parallel Concurrency in typical production run	10k	100k
Wall Hours per Run	12-24	12-24
Aggregate Memory	5 TB	5-10 TB
Memory per Core	500 MB	100 MB
I/O per Run Needed	1-10 TB	10-30 TB
On-Line Storage Needed	10 TB	50 TB
Archival Storage Needed	500 TB	2.5 PB

8.2.3.5 Support Services and Software

Nothing contributed.

8.2.3.6 Emerging HPC Architectures and Programming Models

Nothing contributed.

8.2.4 Non-Intrusive Uncertainty Quantification for Multi-physics Models

PI: Charles Tong (Lawrence Livermore National Laboratory)

8.2.4.1 Summary and Scientific Objectives

The goal of this work is to enhance the predictive capability of our current DOE ASC multi-physics models through rigorous uncertainty quantification and sensitivity analysis. These models can be characterized by high dimensionality (large number of uncertain parameters), expensive simulation times, nonlinear input-output relationship, the presence of structural uncertainties, and the availability of data at various physics modules but data scarcity at the full system level. The full system includes physics at different scales interacting in a complex manner. The UQ challenges are to rigorously and accurately quantify the output uncertainties due to the different sources of uncertainties and experimental data. Current endeavors to improve predictability at subsystem level using simplified models with medium resolution are already consuming many million CPU hours. Advances in UQ technology and HPC will help investigate simultaneously multiple higher resolution models more accurately in an effort to increase confidence in our physics models.

8.2.4.2 Methods of Solution

A comprehensive UQ approach consists of several steps such as model setup, initial parameter selection, parameter distribution prescription, data integration, dimension

reduction, response surface analysis, uncertainty and quantitative sensitivity analysis, model validation, and peer review/documentation. Many of these require large ensemble calculations. The size of the ensemble runs depends on the number of parameters, model nonlinearities, and the number of models under simultaneous study. For example, to perform response surface analysis, an initial sample is generated. HPC resources are then requested to perform the simulation runs. The runs can be submitted collectively or independently. Some of the runs may fail due to system failure or code failures. These failures have to be pruned and the rest are post-processed to extract the output quantity of interest. These results are then analyzed to decide the sample size used is enough to give sufficiently accurate surrogates. This will continue in an iterative manner until some convergence criterion has been satisfied. More sample points are then created and run.

8.2.4.3 HPC Requirements

The models are currently run on classified Linux systems at LLNL. The system currently has 32K nodes each has 4 cores and 4 GBytes of memory. A single simulation run for the milestone completed in 2010 requires approximately 12 hours on 128 processors. Three systems were studied together in the milestone to study predictability and uncertainties using common physics models. The mesh resolution is acceptable for now but is expected to be finer in the future. The total number of runs required to complete the milestone is around 10,000. The milestone could have benefited from even more number of runs.

HPC requirements will be higher in upcoming years due to the need to include more systems into a single UQ study and the need to more accurately characterize uncertainties. Since each sample points can be simulated independently from each other, the availability of more and faster computing nodes will help speed up the study. In addition, faster I/O (especially 'I') will help reduce the simulation time since some large external libraries are read several times during the course of the simulation. Furthermore, more advance job submission systems are needed to handle the thousands of jobs in a seamless manner. Currently, our UQ community uses two modes of job submission. The first is to request a large amount of resources for a period of time and users schedule the runs on their own. The other is to submit a number of jobs at a time and let the job submission system handle the scheduling. To support large ensemble calculations, more thoughts should be given to how best to assist users in their scheduling. Also, for such large number of runs, good support for fault detection and recovery are needed.

8.2.4.4 Computational and Storage Requirements Summary

An emphasis on Uncertainty Quantification (UQ) is a relatively new development for the Office of Science, but it is on the critical path in ASCR's path to exascale. As the Office of Science's investment in computational resources grows, the need to validate and very simulation results becomes more important. The proportion of computer time that will be spent on UQ is uncertain, but a reasonable estimate is that ten percent of the 2013 NERSC ASCR allocation will be needed for UQ runs.

	Current (2010 NERSC)	2014 Requirement
Computational Hours		10% above hours needed for scientific discovery simulations
Parallel Concurrency		No additional requirements
Wall Hours per Run		No additional requirements
Aggregate Memory		No additional requirements
Memory per Core		No additional requirements
I/O per Run Needed		No additional requirements
On-Line Storage Needed		No additional requirements
Data Transfer Needed		No additional requirements

8.2.4.5 Emerging HPC Architectures and Programming Models

Since non-intrusive UQ treats simulation codes as "black boxes", the impact of many-core hardware architectures and/or architectures with heterogeneous nodes on how to perform UQ is little. However, in the future when more intrusive UQ methods are practical, these issues will become very important.

9 Visual Analytics and Data Management

9.1 Visual Analytics and Data Management Overview

With the anticipated improvements in both experimental and computational capabilities, the amount of data (from observations, experiments, and simulations) will be unprecedented.

For example, by 2014 fusion simulations will use 1 billion cells and 1 trillion particles. Based on mean-time-between-failure concerns when running on a million cores, these codes will need to output 2 GBs/sec per core or 2 PB/sec of checkpoint data every 10 minutes. This amounts to an unprecedented input/output rate of 3.5 terabytes/second. The data questions to consider at the extreme scale fall into two main categories: data generated and collected during the production phase, and data that need to be accessed during the analysis phase.

Another example is from climate modeling where, based on current growth rates, data sets will be hundreds of exabytes by 2020. To provide the international climate community with convenient access to data and to maximize scientific productivity, data will need to be replicated and cached at multiple locations around the globe.

These examples illustrate the urgent need to refine and develop methods and technologies to *move, store,* and *understand* data.

The data issue is cuts across all fields of science and all DOE Office of Science Program Offices. Currently, each research program has its own data-related portfolio; ASCR program managers envision an integrated data analytics and management program that will bring multi-disciplinary solutions to many of the issues encountered in dealing with scientific data.

In Applied Mathematics Research, data analytic needs include

- Improved methods for data and dimension reduction to extract pertinent subsets, features of interest, or low-dimensional patterns, from large raw data sets;
- Better understanding of uncertainty, especially in messy and incomplete data sets; and
- The ability to identify, in real time, anomalies in streaming and evolving data is needed in order to detect and respond to phenomena that are either short-lived or urgent.

In Computer Science Research, issues being examined include

- Extreme-scale data storage and access systems for scientific computing that

minimize the need for scientists to have detailed knowledge of system hardware and operating systems;
- Scalable data triage, summarization, and analysis methods and tools for in-situ data reduction and/or analysis of massive multivariate data sets;
- Semantic integration of heterogeneous scientific data sets;
- Data mining, automated machine reasoning, and knowledge representation methods and tools that support automated analysis and integration of large scientific data sets, especially those that include tensor flow fields; and
- Multi-user visual analysis of extreme-scale scientific data, including methods and tools for interactive visual steering of computational processes.

Next-generation Networking Research is concerned with

- Deploying high-speed networks for effective and easy data transport;
- Developing real-time network monitoring tools to maximize throughput; and
- Managing collections of extreme scale data across a distributed network

9.2 Visual Analytics and Data Management Case Studies

9.2.1 Data Analytics and Visualization

Principal Investigator: E. Wes Bethel (Lawrence Berkeley National Laboratory)
Contributors: Prabhat and H. Childs (Lawrence Berkeley National Laboratory)
NERSC Repository (2011): m636

9.2.1.1 Summary and Scientific Objectives

The Berkeley Lab Visualization Group conducts basic and applied research and development of algorithms and software architectures that enable visual data analysis and exploration of very large and complex scientific data using advanced computational platforms.

There are two fundamental motivations for this work. First, scientific codes are producing data of increasing size, resolution, and complexity. Second, existing software tools and algorithms will likely be incapable of processing such data on future platforms. To address these challenges the technology for enabling scientific insight from such data must evolve to take advantage of emerging computational platforms and new fundamental algorithms built atop emerging software architectures must solve the problem of increasing complexity.

The work represented here spans several different but interrelated projects: (1) the DOE SciDAC Visualization and Analytics Center for Enabling Technology (VACET); (2) an ASCR base program project entitled "High Performance Visualization"; (3) an ASCR 10-256 project focusing on high-performance, parallel I/O having an emphasis on end-to-end visual data analysis application performance; (4) a BER-funded 10-05 project entitled "Visual Data Exploration of Ultra-large Climate Data." A theme common among these projects is the desire to enable use of large-scale computational infrastructure to accelerate scientific knowledge discovery. These projects reflect a portfolio of work that ranges from fundamental algorithmic research to advanced development. Most are multi-institutional, and all involve collaborative relationships with application sciences.

9.2.1.2 Methods of Solution

Many of our projects use the VisIt visualization software application (www.llnl.gov/visit). VisIt is a production-quality, petascale-capable visual data analysis application that has a large, worldwide scientific user community. VisIt is written in C++ and uses MPI. In recent years, we have demonstrated that VisIt can run at very high concurrency on all platforms at DOE supercomputing centers.

We also use, develop, and optimize software libraries for specific types of operations, e.g. parallel I/O. Performing I/O efficiently using 100K+ cores is a huge challenge. We have

been focusing on the optimization of the HDF5 I/O library (and the NetCDF-4 library, which relies on HDF5) for use on DOE's large computational platforms. We are trying to leverage multi-core architectures to accelerate I/O by using advanced features like append-only I/O, asynchronous I/O, elimination of collective metadata operations, auto-tuning for specific file systems, and better resiliency.

Visual analysis is a highly data intensive activity. The I/O to flop ratio is much lower than it is in many simulation codes. As an example consider uncertainty quantification in a climate modeling application. A climate scientist might run multiple atmospheric models varying the initial conditions. Each run would produce many years of simulation data. As a post-processing step, all files from all runs would be read by the visual analysis application to produce a statistical analysis that conveys the amount of variance across the different runs.

Currently, most visual data analysis if performed as a post-processing step after completion of a scientific simulation. In the future we expect more analytics to be performed concurrently with simulations because I/O rates are not keeping up with computational capability and it is already becoming cost-prohibitive to write out all data at full spatiotemporal resolution. VisIt includes a "simulation API" that supports *in-situ* use thereby avoiding intermediate I/O. We do not expect that concurrent, or *in-situ* visual data analysis will ever completely replace the post-processing approach, but *in-situ* analysis will become increasingly important in the future.

9.2.1.3 HPC Requirements

Our research often focuses on designing and testing approaches on problems having size and complexity representative of future scientific workloads. Our runs require access to a wide range of machine configurations, including full systems. For example, we conducted 32K-core runs on Franklin in 2010; these may have been some of the first of that size on that machine. This also implies a requirement for a large allocation. Optimization and performance tuning, an important part of the process here, consumes a lot of resources. For example, during the period January 2010 through May 2010, our team used over 7M hours at the NSF's National Institute for Computational Sciences machines. The runs at NICS, part of a Director's discretionary allocation, exceeded our allocation at NERSC by a factor of seven.

Nearly all of our work requires a high-performance I/O subsystem. We can reasonably expect that by 2014, I/O test runs will routinely move tens to hundreds of TB into or out of the machine per run.

We run both surrogate applications (like IOR for I/O benchmarking), and actual science codes (such as Impact-T and GCRM).

Visual analysis algorithms are diverse in their underlying mathematical models and in how they access memory. Some algorithms use a well-ordered, structured memory access

pattern while others use an unstructured memory access pattern (e.g., raycasting volume rendering, where each ray computation requires random access to a large, in-memory dataset). Memory requirements are diverse as well, but analysis algorithms may be memory intensive and can benefit greatly from up to 8 GB per core.

By 2014 we will need to deal with 100s of TB per run. At present, our maximum-sized datasets are on the order of 10s of TBs. This is due to a combination of the relatively slow I/O speed on existing computational platforms and limitations of filesystem space allocations. Our software tools can easily load and process more data than 10s of TB. If we try to load a 1 TB dataset over a 10GB/sec link, our code requires approximately 100 seconds to perform I/O. With future simulations producing 100s of TBs, we will need I/O bandwidth to increase by factors of 10s to 100 to local storage.

9.2.1.4 Support Services and Software

We often require interactions with NERSC user support to find the right combinations of environment variables needed for jobs at very high level of concurrency.

Beyond the usual collection of software/services needed to support designing/running large-concurrency visual analysis jobs on parallel platforms, our team makes extensive use of several other types of software/services, much of which is presently provided by local IT infrastructure, but that could arguably be better provided by centralized facilities like NERSC. We would like "one-stop shopping" for project-wide collaborative software infrastructure: wikis; web-hosting; remote access to SVN/CVS configurable by the PI or Proxy to specific users in specific ways; email lists, etc. Presently, NERSC hosts the web site www.vacet.org. The only improvement we can suggest here is for some type of SCM system that would allow us to designate various team members with privileges to update content. (Presently, we route all content changes through one person, a system administrator in Utah.) We make extensive use of CVS for intra-team document writing, and presently use https://codeforge.lbl.gov/, which is provided by HPCRD. Some projects that are sufficiently large, like VisIt, have their SVN repository hosted by NERSC. Perhaps something like codeforge for NERSC users might make sense: it would streamline our operations somewhat in that those team members who have a NERSC login could benefit from these additional services, which are a crucial part of our project's operations.

We need performance data collection and analysis tools to better understand how data movement through the memory hierarchy affects performance and scalability. In the future the "memory hierarchy" will extend out to file systems for I/O projects.

9.2.1.5 Computational and Storage Requirements Summary

	Current (NERSC 2010)	2014 Requirement
Main Science Driver	CS research on algorithms and architectures to enable scientific insight.	
Computational Hours	1.4 M	500M
Parallel Concurrency	1K-216K	1K-10M
Wall Hours per Run	Minutes to an hour	Minutes to an hour
Aggregate Memory	1-300TB	1-30PB
Memory per Core	1-2GB	1-8GB
I/O per Run	10s of TB	100s of TB
On-Line Storage Needed	10s of TB	100s of TB
Data Transfer	Local: 10GB/s Remote: 10MB/s	Local: 100-500GB/s Remote: 100MB/s
Archival Storage	10s of TB	100s of TB

9.2.1.6 Emerging HPC Architectures and Programming Models

In the past couple of years, our team has had remarkable success in R&D of hybrid-parallelism for visual data analysis. We have combined MPI+pthreads, MPI+OpenMP, and MPI+CUDA to produce a hybrid-parallel raycasting volume renderer that has been run 216K-way parallel on 23000^3 grids (which uses 300TB of aggregate memory) on JaguarPF, the highest level of concurrency ever published in our field.

Like others in the field, we are keen to take full advantage of these new processors, and we believe that using them will require a blend of MPI and some underlying shared-memory, data-parallel form of design/coding. Our team has developed custom hybrid-parallelization approaches for difficult visualization problems, like flow-field visualization, where we use a novel, hybrid parallel approach with a dynamic blend of problem decomposition strategies to achieve both good load balance and scalability. These are C/C++ codes that use MPI and also invoke shared-memory parallel code inside a socket to take advantage of multi-/many-core, shared-memory architectures. We have explored combining MPI with OpenMP, POSIX threads, and CUDA. This research accounts for the most intensive use of computational resources (to date) for our projects. We also need access to advanced emerging, professionally operated computational platforms and infrastructure, especially "testbeds" containing emerging computational architectures that might be too cost-prohibitive for a single research project.

Doing this work requires support for the latest versions of CUDA, OpenCL, etc., and a diversity of test platforms of sufficient size for meaningful performance/scalability runs.

We are concerned about the scalability of Pthreads and OpenMP as the core count on processors grows higher. On the other hand, CUDA is an NVIDIA-only product and as such may likely not be generally available for use on multi-core CPUs. For an academic study, it may be fine to limit oneself to a proprietary technology, but for production

quality software that will be widely distributed, we are hoping for a solution based upon an industry standard for which there is broad support, e.g. OpenCL. Another area of concern would be if future multi-/many-core processors do not support a shared memory space. In other words, if future m-core processors don't support shared memory, then it may be difficult to realize good scalability and performance on those platforms.

9.2.2 Visualization and Analysis of Volume and Particle Data from Turbulent Combustion Simulations

PI: Kwan-Liu Ma (University of California, Davis)
Contributors: Jackie Chen, Ray Grout, Hongfeng Yu (Sandia National Laboratories); Jishang Wei (University of California, Davis)

9.2.2.1 Summary and Scientific Objectives

With the rising power of supercomputers, state-of-the-art combustion simulations can begin to reliably predict efficiency and pollutant emission for new engines and new fuels. The goal of this work is to support the data visualization and analysis needs of scientists who are developing and conducting predictive simulations with sufficiently high fidelity and physical complexity. The new data understanding strategies we have been developing allow our collaborators to capture and see previously unseen information from their simulation data at unprecedented quality and detail. This work has been sponsored in part by a DOE SciDAC-2 Ultrascale Visualization grant and an NSF PetaApps grant.

The fundamental problem to address is the growing scale of the data generated by the turbulent combustion simulations routinely performed by our collaborators at the Sandia National Laboratories, Oak Ridge National Laboratory, and University of Michigan. How do we make it easy for them to extract and study important information from the hundreds of terabytes to petabytes of data that their simulations are capable of outputting today and the exabytes that will be output in the not very distant future? Essentially, from saving, moving, to processing the data for validation and analysis, we must have scalable solutions. Our study thus involves experimenting with various strategies we have been developing in a realistic supercomputing environment. Our findings will also help our collaborators to better estimate their computing requirements in the future.

9.2.2.2 Methods of Solution

We are developing both *in-situ* and post-processing visualization and analysis methods. With the *in-situ* approach, we consider direct rendering of multivariate data, feature extraction and tracking, and data reduction tasks. In the post-processing approach, we focus on the development of novel particle data analysis methods and their acceleration for high performance and interactive exploration.

In-situ rendering of the multivariate field data may be used to create an animation of the whole simulation. Note that for the turbulent combustion simulations under consideration, our collaborators can only afford to store a small fraction of the hundred thousand time steps of their data. That is, substantial amounts of data are discarded to cope with the limited storage space provided to them. As a result, complete data are only available at the simulation time, and in-situ visualization can thus capture certain aspects of the modeled phenomena that are essentially lost by the time post-processing occurs.

That is, we can also consider in-situ visualization as a good way to provide a view of the discarded data.

The combustion simulations also keep track of millions to billions of transported particles throughout the simulated flames. These particles capture the dynamic behavior of the modeled flames, and provide a Lagrangian description of the combustion environment. We aim to develop an interactive particle querying strategy, ideally for a desktop setting, based on both in-situ and post-processing visualizations of particle trajectories. Our collaborators want to test hypothesis about the nature of the particle trajectories. To such tasks, the large amounts of particle histories need to be categorized, which is generally done with clustering. However, parallel clustering at scale remains a challenging research topic.

Key research tasks going forward include:

- Development/optimization of parallel analysis algorithms for clustering, topological feature extraction, feature tracking, etc.;
- Development of *in-situ* data reduction methods for both field and particle data;
- Development of remote visualization support.

9.2.2.3 HPC Requirements

The *in-situ* visualization code is usually tightly coupled with the combustion simulation code, which presently runs on the Cray XT5 at ORNL using up to 48,000 cores taking as much as 14 million CPU hours, while the post-processing code runs on a GPU cluster. Our collaborators store up to 300 TBs of field data, and 10 TBs of particle data for each run of the simulation. Our current in-situ visualization computational cost is limited to 5 percent or less of the overall supercomputing time used by the simulation. This is quite acceptable for our collaborators. Particle data visualization is exclusively done in post-processing steps, but in the future as much calculation as possible will be carried out *in situ*. Our goal, through, is *in-situ* processing, to keep the growth rate of the data that the scientists need to look at more in line with that of the memory size and network bandwidth.

This project was not allocated resources at NERSC in 2010, but according to our collaborators who do have NERSC projects, in 2011, simulations are expected to use from 90,000 to 120,000 cores on Jaguar or Hopper. Field data resolution will grow to 7 billion grid points, but particle data will remain the current size, which is about 50 million. Our collaborators expect that these numbers will grow linearly with the size of the machines through 2014. This also suggests that they will be able to fully utilize what NERSC can provide in terms of both the computing power and storage space. For post-processing visualization and analysis tasks, a 64-node GPU cluster (with up to 2GB video memory per card) connected to the data servers with networks at 100Gb/sec will be sufficient.

By 2014, we will need to better support our collaborators' needs to study particle trajectory data. We envision this will require additional hours and cores per simulation to accommodate the added in-situ particle data preprocessing calculations. It is clear to us that that in order not to hold up the simulation, we should move the particle data to a separate set of processing nodes allocated to the complex calculations. This increase in allocation is estimated to be 15%.

9.2.2.4 Support Services and Software

The new capabilities that we aim to provide to our science collaborators cannot be built on top of existing visualization toolkits. Our software libraries and systems are mostly implemented in C++, MPI, and CUDA. For providing remote visualization services, we will want to make use of what NERSC has already created for both web-based interfaces and data movement.

9.2.2.5 Emerging HPC Architectures and Programming Models

No info provided.

10 Math Software

10.1 Math Software Overview

Math software lies at the heart of every science and engineering application; it is paramount that this software is of the highest quality, embodies the most advanced numerical algorithms, and is useful and usable by application scientists.

Math software is also where mathematics and computer science intersect: it is not enough to simply provide good mathematical algorithms; the software also needs to run on today's and tomorrow's architectures. Computer science issues such as programming models, data movement, I/O, storage, systems software all come into play.

ASCR sponsors numerous NERSC users who develop and test mathematical software; most have small (< 100,000 processor hour) allocations but tend to use many processors. The major computational challenge for mathematical software is scalability.

Exascale computing will only add to the complexity of math software. Will these codes supported by DOE survive the paradigm shift? Again, data movement, communication, and memory management will be the key to scalability. No easy solution is in sight, but it is expected that the boundary of mathematics and computer science will be blurred as software developers tackle the issues of making the codes architecture aware, reducing communication in the algorithms, re-implementing algorithms to swap flops for data locality, adding resilience and fault tolerance, and reducing energy consumption.

10.2 Math Software Case Studies

10.2.1 High Performance Sparse Matrix Algorithms

PI: Esmond G. Ng (Lawrence Berkeley National Laboratory)
Contributors: Xiaoye Sherry Li, Chao Yang (Lawrence Berkeley National Laboratory)
NERSC Repository: mp127

10.2.1.1 Summary and Scientific Objectives

The goal of this project is to design and implement highly efficient and scalable algorithms for solving sparse matrix problems on various classes of high-performance computer architectures. Problems to be considered include, but are not limited to, direct

methods and iterative methods for solving sparse linear systems, preconditioning techniques for iterative methods, and sparse eigenvalue problems. For sparse linear systems, our focus is on techniques for handling highly indefinite and ill-conditioned matrix problems. For eigenvalue calculations, we are interested in both linear and nonlinear problems. We also investigate numerical optimization techniques, since they often require sparse matrix technology. Our work is tied closely with scientific applications in the SciDAC program.

10.2.1.2 Methods of Solution

This project is concerned with the development of advanced numerical algorithms for solving sparse matrix problems. We design and develop codes for solving sparse systems of linear equations. Our codes include SuperLU and PDSLin. SuperLU is based on triangular factorization. PDSLin uses a hybrid approach, which applies triangular factorization to a portion of the matrix and uses iterative methods on the rest of the matrix. Our work on preconditioning techniques is based on incomplete factorizations. We are also working on the solution of sparse eigenvalue problems, in which case PARPACK is our primary code.

Parallelism in all our codes is expressed using MPI.

10.2.1.3 Computational and Storage Requirements Summary

	10.2.1.3.1 Current (NERSC 2010)	2014 Requirement
Computational Hours	400,000	2 M
Parallel Concurrency	64-8,000	2,500 to 20,000+
Wall Hours per Run	Up to 3 hours	5+ hours
Aggregate Memory	Can use all available memory	Can use all available memory
Memory per Core	Can use all	Can use all
I/O per Run Needed	40 GB, mostly problem input	Probably much larger due to increased problem size

10.2.1.4 Support Services and Software

We need ParMETIS and PTScotch for graph partitioning, PETSc for iterative solvers, and BLAS, LAPACK, and SLAPACK for handling dense kernels.

10.2.1.5 Emerging HPC Architectures and Programming Models

We are beginning to investigate the use of hybrid programming paradigms in some of our sparse matrix algorithms. Algorithmic changes are expected, including possibly the

incorporation of dynamic load balancing techniques, when/if the number of cores increases on a node.

10.2.2 DOE Advanced CompuTational Software Collection

Principal Investigator: Tony Drummond (Lawrence Berkeley National Laboratory)
Contributor: Osni Marques (Lawrence Berkeley National Laboratory)
NERSC Repository: m340

10.2.2.1 Summary and Scientific Objectives

The Advanced Computational Software Collection (ACTS) is a set of DOE-developed libraries and software tools that make it easier to develop high performance applications. These tools can be used to solve problems that are common to many engineering and scientific applications.

This project uses NERSC resources to perform the installation and testing of the libraries and software tools, as a mechanism for the proper functioning of applications that are based on those tools. The project also makes the tools available to NERSC users and provides support when questions and problems arise.

10.2.2.2 Methods of Solution

ACTS tools provide a range of efficient numerical algorithms as well as support for code development, execution and optimization. The numerical algorithms include direct methods for the solution of linear systems of equations and eigenvalue problems, (Krylov-type) iterative methods for the solution of linear systems of equations and eigenvalue problems, nonlinear solvers, and nonlinear optimization solvers. Algorithmic implementations are (mostly) MPI-based.

10.2.2.3 HPC Requirements

Typically, installing, updating, and testing the set of tools take a few thousand MPP hours. Testing is performed on a few hundreds to a few thousands of cores. This is done on all MPP machines supported at NERSC. It is expected that highly optimized BLAS are available on these machines.

By 2014 we expect to focus (i.e. test and incorporate in the ACTS Collection) new algorithmic implementations that are optimized for upcoming, hybrid architectures. These may imply implementations based on hybrid programming models; and tests will likely have to be carried out using high levels of concurrency. Access to prototype architectures will very likely be necessary.

10.2.2.4 Computational and Storage Requirements Summary

	Current (2010 NERSC)	2014 Requirement
Computational Hours	20 K	250 K
Parallel Concurrency	1,00s of cores	1,000s of cores
Wall Hours per Run	1 hour	1 hour

10.2.2.5 Support Services and Software

PAPI and highly optimized BLAS are required. If a tool needs a specific library we provide the library with the tool.

10.2.2.6 Emerging HPC Architectures and Programming Models

Our plan to accommodate many-core hardware architectures and/or hybrid architectures consists of evaluating the developments of projects funded by ASCR and including those in the funding opportunities via Joint Mathematics/Computer Science Institutes, X-Stack Software Research, Exascale Co-Design Centers, and SciDAC.

11 Computer Science and Performance Evaluation

11.1 Computer Science and Performance Evaluation Overview

Understanding the performance of a code on a particular computer paves the way for increased efficiency and algorithmic innovations. Often it is not straightforward for the application code users or developers to discover where and how the improvements can be made. Thus performance engineering has played an important role in the advancement of high-performance computing. As supercomputers become more complex, application developers and users will increasingly rely on computer scientists to provide the tools and the knowledge to maximize performance.

Going forward, critical computer science challenges for exascale computing includes

- Communication
- Energy efficiency and power management
- Fault tolerance and resiliency
- Data locality
- Memory hierarchy
- Resource scheduling

Current ASCR investments in areas relevant to performance evaluation include a number of Joint Math/CS Institute projects, X-Stack software research, and the SciDAC Institute project SUPER. NERSC machines are used to study performance and tuning issues. Most of these projects have small allocations but use many processors (16K and more).

11.2 Computer Science and Performance Evaluation Case Studies

11.2.1 Advanced HPC Programming Technologies Center

Principal Investigator: John Shalf (Lawrence Berkeley National Laboratory)
Contributors: Nicholas Wright, Hemant Shukla, and David Donofrio (Lawrence Berkeley National Laboratory)

11.2.1.1 Summary and Scientific Objectives

The goal of this project is to better understand the requirements of the DOE/NERSC workload, assess emerging system technologies, and use the workload requirements to

drive changes in computing architecture that will result in better HPC system architectures for scientific computing in future generation machines. One of the primary tasks is to continually assess available HPC system solutions using a combination of application benchmarks and microbenchmarks.

Specific areas we are targeting in the next few years are related to the Combustion Exascale co-design Center, the CoDesign for Exascale (CoDex) project, and the Exascale Executions models project. These all involve work to understand the performance characteristics of scientific applications, with a view to achieving substantially increased performance on future architectures through the identification of performance issues on today's architectures and the co-design of the hardware and software of the future.

11.2.1.2 Methods of Solution

This project is concerned with the evaluation of current technologies with a view to the design of future ones. In order to do this we will use a variety of performance measurement tools to collect measurements. We will also perform a number of performance scaling and optimization experiments. The combustion co-design center will focus on the combustion codes S3D and LMC. The execution models work will also look at these codes as well as GTC.

Parallelism in all these codes is expressed using MPI and OpenMP currently. In the future we expect to examine PGAS performance as well as other parallelization paradigms as and when required.

11.2.1.3 HPC Requirements

In order to develop new computational algorithms with increased performance we are starting to require more and more computational resources. This is for two reasons: the complexity of the optimization space we are exploring is growing, and there will be an increased need to explore new techniques because of the changes coming in the Exascale era. For example, the PGAS implementation of the GTS fusion code used 2.5 M core hours to explore different algorithms at scale.

By 2014 we expect that we will need approximately 35 Million core hours. The increased concurrencies of machines in that time frame provide approximately a factor of four increase, with the remaining amount coming the higher number of performance measurement and optimization experiments expected to be required in this timeframe.

Key science tasks going forward include: the identification of the deficiencies in current execution models and the design of new ones, continuing our performance and benchmarking studies, and the development of new mechanisms for the collection and processing of performance data.

11.2.1.4 Computational and Storage Requirements Summary

	NERSC 2010	2014
Main Science Driver	Various scientific applications	Various scientific applications
Computational Hours	5.6 M	35 M
Parallel Concurrency	256-100,000	2,500 to over 200,000
Wall Hours per Run	Up to 1 hour	Up to 1 hour
Aggregate Memory	Can use all available memory	Can use all available memory
Memory per Core	Can use all	Can use all
I/O per Run	10-100GB, (Performance traces can be large)	100GB-10 TB (Performance Tracing data)

11.2.1.5 Support Services and Software Required

We need performance analysis tools such as IPM, CrayPat and HPCToolkit. As much performance information as the runtime and scheduler can provide is very useful. Things such as: task placement within the machine, access to hardware counters (both in the processor and network chip) and measurements of energy usage. Related to this is good documentation from the vendor(s), making a measurement of an ill-defined property of the machine can provide misleading information.

11.2.1.6 Emerging HPC Architectures and Programming Models

Our research is focused upon these topics. Therefore access to early hardware prototypes is crucial. Also useful would be a software environment that allows us to experiment with modifications to the runtime system to explore the potential optimizations and modifications to the execution model if required would be useful.

11.2.2 UPC, CAF, and Titanium

Principal Investigator: Kathy Yelick (Lawrence Berkeley National Laboratory)
NERSC Repository: mp215

11.2.2.1 Summary and Scientific Objectives

MPI is currently the de facto programming interface for inter-process communication on the world's largest high performance computers, but "partitioned global address space" (PGAS) languages offer advantages in both programmability and performance relative to MPI. This project's goal is to develop, test, and implement compilers and software infrastructure that will enable developers to use PGAS languages to increase both the usability and efficiency of high-end machines. PGAS allows for efficient one-sided communication on clusters, and shared memory features on multicore and SMP nodes. By providing a global address space, they permit application software to directly share data when appropriate, rather than partitioning and replicating it. These languages have been supported by several federal agencies, including DOE, DARPA, NSA, and NSF.

UPC is an explicitly parallel language, based on the C language. Co-Array Fortran is part of the Fortran 2008 language specification. Titanium is based on Java. UPC allows the programmer to do optimizations by hand, e.g., through the use of non-blocking data movement functions for bulk data movement. Our UPC application work includes a fast 3D FFT solver that overlaps communication with computation, a dense LU factorization code, and a sparse Cholesky factorization code. Applications in Titanium include a heart simulation developed using NSF funding and an adaptive mesh refinement solver developed by Phil Colella's group at Berkeley Lab.

Specific goals of the project include:

1. Make the languages available on all high-end platforms. While some vendors (HP, SGI and Cray) provide UPC implementations for their large machines, the Berkeley UPC and Titanium compilers developed by this project run portably across these and other architectures.
2. Provide an efficient and portable communication layer, GASNet, for all global address space languages. GASNet is used by the Berkeley UPC compiler, the Cray compiler for Co-Array Fortran and UPC compilers (on its SeaStar network), the Intrepid gcc-based UPC compiler, the Rice Co-Array Fortran compiler, and the Berkeley Titanium compiler. It is also being used for implementing the Cray Chapel language developed as part of the DARPA HPCS effort.
3. Demonstrate the effectiveness of these languages on real applications that can produce scientific results not possible with other codes. Explore language extensions motivated by specific application demands.
4. Evaluate the performance of these languages by comparing to other library-based models like MPI on both microbenchmarks and application level benchmarks.

5. Develop compiler analyses and optimizations that automate some of the performance-tuning process.

11.2.2.2 Methods of Solution

Since this is computer science research to develop tools used by others, there is no fixed science problem to be run. Work in 2010 focused on optimizing the multicore support for these languages, including use with a hybrid runtime that mixes shared memory and networked cores.

11.2.2.3 HPC Requirements

The Berkeley group works on its implementation of UPC and the underlying GASnet communication layer used by Cray UPC/CAF/Chapel on its large systems and by the Intrepid (gcc) UPC implementation, in addition to the Berkeley UPC and Titanium implementations. NERSC systems are used for testing an scalability tuning of the Berkeley and Rice compilers and runtime systems. The group has developed an optimized implementation of the basic GASNet communication layer for Petascale systems. The team builds on this work by implementing and evaluating application benchmarks with large-scale runs using 10s of thousands of cores. Tests on Cray XT4s and IBM Blue Gene/P show that the UPC implementation outperforms MPI code, with a gap that grows for larger core counts.

We have also been developing highly optimized and scalable collective implementations and showed that properties, such as the best choice of tree structure and communication strategy is highly dependent on the machine size, collective routine, and data size. This produces an enormous search space.

Research at Rice focused on research, development, and evaluation of a compiler and runtime system for Co-array Fortran (CAF) 2.0. To assess the utility of Coarray Fortran 2.0 for developing scalable parallel applications for distributed-memory platforms such as the Cray XT and the Blue Gene platforms, we developed implementations of the HPC Challenge benchmarks: STREAM, HPL, RandomAccess, and FFT. We will continue to develop and test these benchmarks at scale.

Efficient development and implementation of runtime environments for PGAS languages requires access to low-level messaging interface, e.g. the Cray Gemini software stack.

11.2.2.4 Computational and Storage Requirements Summary

	Current NERSC (2010)	2014 Requirement
Computational Hours	388 K	10 M
Parallel Concurrency	4,096	16K

11.2.2.5 Support Services and Software

We need to have the following software available: C, C++, MPI, LAPI (IBM), PThreads, FFTW, System V shared memory, Chombo, LAPACK, Metis

Appendix A. Attendee Biographies

John Bell is a Senior Staff Mathematician at Lawrence Berkeley National Laboratory and leader of the Center for Computational Sciences and Engineering in LBNL's Computational Research Division. Prior to joining LBNL, he held research positions at Lawrence Livermore National Laboratory, Exxon Production Research and the Naval Surface Weapons Center. John's research focuses on the development and analysis of numerical methods for partial differential equations arising in science and engineering. He has made contributions in the areas of finite difference methods for hyperbolic conservation laws and low Mach number flows, discretization strategies for multiphysics applications, and parallel computing. He has also pioneered the development of adaptive mesh algorithms for multiphysics and multiscale problems. John's work has been applied in a broad range of fields, including aerodynamics, shock physics, seismology, flow in porous media and astrophysics.

Wes Bethel is a Staff Scientist at Lawrence Berkeley National Laboratory, where he is a member of and Group Leader for the Scientific Visualization group. The group's activities include visualization research and production visualization/analytics, both of which focus on increasing scientific productivity through better visual data understanding technologies. Bethel's research interests include computer graphics and visualization software architecture, remote and distributed visualization algorithms, latency tolerant and parallel graphics techniques. Bethel received an MS in Computer Science from the University of Tulsa in 1986, a PhD in Computer Science from the University of California – Davis in 2010, and is a member of ACM, ACM/SIGGRAPH and IEEE.

Alok Choudhary is John G. Searle Professor of Electrical Engineering and Computer Science and a Professor of at Kellogg School of Management at Northwestern University. He is the founding director of the Center for Ultra-scale Computing and Information Security (CUCIS), which involves several schools, National Labs and universities. He is the academic director of the Executive Program on "Managing Customer Relationships for Profit" in the Kellogg School of Management's executive education program, and is a member of the Center for Genetic Medicine.

Johan Larsson is a Research Associate at the Center for Turbulence Research at Stanford University. His research interests are in Turbulent combustion, Shock/turbulence interaction, Near-wall modeling for large eddy simulation, Mixing of passive scalars with application to combustion, Numerical analysis, Large-scale parallel computation, and Aeroacoustics. He holds a Ph.D. in Mechanical Engineering from the University of Waterloo, Canada.

Sanjiva Lele has a joint appointment in the Department of Mechanical Engineering and in the Department of Aeronautics and Astronautics at Stanford University. Professor Lele's research combines numerical simulations with analytical modeling to study fundamental unsteady flow phenomena, turbulence, flow instabilities, and flow-generated sound. Recent projects include simulation and modeling of high-speed jets and shock-cell noise, exploitation of flow instabilities for enhanced mixing and for reducing the vortex-wake hazard from an airplane, new approaches for active/passive noise control, and the development of high-fidelity prediction methods for engineering applications including transition and flow-generated noise. Professor Lele has been an Associate Editor of the Journal of Fluid Mechanics since 1994.

Kwan-Liu Ma is a professor of computer science and the chair of the Graduate Group in Computer Science (GGCS) at the University of California-Davis. He leads the VIDI (Visualization and Interface Design Innovation) research group, and directs the DOE SciDAC Institute for Ultra-Scale Visualization. His research spans the fields of visualization, high-performance computing, and user interface design. Professor Ma received his PhD in computer science from the University of Utah in 1993. During 1993-1999, he was with ICASE/NASA Langley Research Center as a research scientist. He joined UC Davis in 1999.

Osni Marques is a staff scientist at the Scientific Computing Group, Lawrence Berkeley National Laboratory (LBNL). One of his projects consists in making advanced software tools more widely used and more effective in solving DOE's and the nation's scientific problems. He has studied and implemented algorithms for the solution of problems in numerical linear algebra, in particular eigenvalue problems, that have been used in applications related to protein motions, acoustics problems in automobile design, structural analyses, and also inverse problems. He is also a member of the Sca/LAPACK development team.

Dan Martin is a Computational Scientist at Lawrence Berkeley National Laboratory.

Esmond Ng is a Senior Scientist at Lawrence Berkeley National Laboratory and head of the Applied Mathematics and Scientific Computing Department in LBNL's Computational Research Division. Prior to joining LBNL, he was a Senior Staff Member at Oak Ridge National Laboratory. Esmond's research interests include sparse matrix computation, numerical linear algebra, parallel computing, computational complexity, and mathematical software. He has been a key contributor in the DOE SciDAC program and working closely with application scientists. Esmond received his Ph.D. in Computer Science from the University of Waterloo.

Karen Pao is the Program Manager for ASCR's applied math base programs. Karen worked at the Los Alamos National Laboratory (LANL) for nearly 20 years. Karen's career at LANL included an early-career exposure to high-performance computing and benchmarking, experimentation with C++ array class libraries, designing numerical schemes for slightly compressible flows, and, most recently, performing simulations and analyses of underground nuclear test output in the Hallowed Halls of the famous Top-

Secret X-Division. She was on assignment with the Advanced Simulation and Computing (ASC) Program at the Department of Energy/National Nuclear Security Administration Headquarters (DOE/NNSA HQ), where her primary assignment was to devise the ASC National Verification & Validation Strategy, when she decided to stay in Washington, DC for good. Karen received her Ph.D. in Mathematics from UCLA in 1993.

Rob Ross of Argonne National Laboratory is a pioneer in the design of parallel file systems and high-performance interfaces for managing large datasets. He led the development of the Parallel Virtual File System (PVFS) used in many academic, industry, and laboratory settings, including the Nation's leadership-class computing facilities. He leads storage research in the DOE SciDAC Enabling Technology Center for Scientific Data Management and is Associate Director of the SciDAC Institute for Ultra-Scale Visualization, where he is developing tools to help researchers address challenges in storage, retrieval, and the extraction of meaning from very large scientific datasets. He is also a Fellow of the University of Chicago/Argonne Computation Institute.

Nagiza Samatova is Associate Professor of Computer Science at North Carolina State University. She received the B.S. degree in applied mathematics from Tashkent State University, Uzbekistan, in 1991 and her Ph.D. degree in mathematics from the Russian Academy of Sciences, Moscow, in 1993. She also obtained an M.S. degree in Computer Science in 1998 from the University of Tennessee, Knoxville. Dr. Samatova specializes in computational biology and high performance data mining, knowledge discovery and statistical data analysis. She is the author of over 50 publications, one book, and two patents.

Arie Shoshani is a senior staff scientist at Lawrence Berkeley National Laboratory. He joined LBNL in 1976. He heads the Scientific Data Management Group. He received his Ph.D. from Princeton University in 1969. His current areas of work include data models, query languages, temporal data, statistical and scientific database management, storage management on tertiary storage, and grid storage middleware. Arie is also the director of the Scientific Data Management (SDM) Center for Enabling Technologies (CET) under the SciDAC-1 and the SciDAC-2 program. In this capacity, he is coordinating the work of collaborators from 5 DOE laboratories and 5 universities (see: http://sdmcenter.lbl.gov). Dr. Shoshani has published over 75 technical papers in refereed journals and conferences, chaired several workshops, conferences, and panels in database management; and served on numerous program committees for various database conferences. He also served as an associate editor for the ACM Transactions on Database Systems. He was elected a member of the VLDB Endowment Board, served as the Publication Board Chairperson for the VLDB Journal, and as the Vice-President of the VLDB Endowment.

Erich Strohmaier is Head of the Future Technology Group of the Computational Research Division at Lawrence Berkeley National Laboratory, USA. His current research focuses on performance characterization, evaluation, modeling, and performance engineering. From 1995-2000, he worked in the Computer Science Department of the University of Tennessee at Knoxville, USA. From 1990-1995, he worked at the

University of Mannheim, Germany, where he was responsible for the Parallel Computing Group. He is, along with Hans Meuer and Jack Dongarra, founder of the TOP500 project.

Charles Tong is a member of the technical staff in the Center for Applied Scientific Computing at LLNL. His current research interests include uncertainty quantification, sensitivity analysis, iterative solvers for linear systems, and parallel computing. He is currently working on several research and application projects on uncertainty quantification.

Richard Gerber is one of the workshop coordinators and report co-editor. He earned a Ph.D. in physics in 1993, specializing in computational astrophysics, from the University of Illinois at Urbana-Champaign. Richard's specialty is N-body and Smoothed Particle Hydrodynamics of colliding galaxies, "ring" galaxies in particular. He held a National Research Council postdoctoral fellowship at NASA Ames Research Center from 1993-1996, working on parallel programming and computational models of galaxies. Richard joined NERSC in 1996. In his position as User Services Consultant, NERSC Training coordinator, and co-facilitator and report editor for these workshops, Richard works with scientists in many fields of physical science and engineering. Richard is responsible for the NERSC help desk portal and has developed NERSC's web interfaces that allow users to query information about running, pending, and completed jobs.

Harvey Wasserman is one of the workshop coordinators and report co-editor. He is in the NERSC User Services Group and has been involved in workload characterization, benchmarking, and system evaluation at NERSC and at Los Alamos National Laboratory for over 24 years.

Appendix B. Workshop Agenda

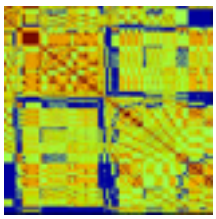
Wednesday, January 5		
Time	Topic	Presenter
8:00am	Arrive, informal discussions	
8:30	Welcome, introductions, workshop goals, charge to committee	Yukiko Sekine, DOE-SC/ASCR
8:50	Workshop outline, logistics, format, procedures	Harvey Wasserman, NERSC
9:00	ASCR Program Office Research Directions	Karen Pao, DOE / ASCR
9:15	NERSC Role in Advanced Scientific Computing Research	Kathy Yelick, NERSC Director
10:10	Break	
10:20	Case Study Introduction: Scientific Data Management	Karen Pao
10:30	Case Study: Scientific Data Management	Arie Shoshani
11:00	Case Study: Future Data Needs	Nagiza Samatova
11:30	Case Study: I/O Software	Alok Choudhary
12:10pm	Working Lunch	
12:20	Case Study Introduction: Visualization	Karen Pao
12:30	Case Study: Visualization	Kwan-Liu Ma
1:00	Case Study: Visualization	Wes Bethel
1:30	Break	
1:50	Case Study Introduction: Applications	Karen Pao
2:00	Case Study: Applications	John Bell
2:30	Case Study: Applications	Charles Tong
3:00	Case Study: Applications	Johan Larsson / Sanjiva Lele
3:30	Break	
3:50	Case Study Introduction: Computer Science & Performance Evaluation	Karen Pao
4:00	Case Study: Computer Science & Performance Evaluation	Erich Strohmaier
4:40	General discussions	
5:00	Adjourn for the day	

Thursday, January 6		
8:00am	Arrive, informal discussions	
8:30	Case Study Introduction: Math Software	Karen Pao
8:40	Case Study: Math Software	Esmond Ng & Osni Marques
9:10	NERSC Initial Summary	Richard Gerber, NERSC
10:00	Break	
10:20	Case study format review; sample case study	Harvey Wasserman, NERSC
10:30	Report schedule and process	Richard Gerber, NERSC
10:45	Q&A, general discussions, breakout sessions	
12:00pm	Workshop ends	

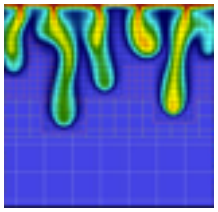
Appendix C. Abbreviations and Acronyms

ALCF	Argonne Leadership Computing Facility
AMR	Adaptive Mesh Refinement
API	Application Programming Interface
ARRA	American Recovery and Reinvestment Act of 2009
ASCR	Advanced Scientific Computing Research
CUDA	Compute Unified Device Architecture
ESnet	DOE's Energy Sciences Network
FFT	Fast Fourier Transform
GPGPU	General Purpose Graphical Processing Unit
GPU	Graphical Processing Unit
HDF	Hierarchical Data Format
HEDP	High Energy Density Physics
HFHI	High Frequency Hybrid Instability
HPC	high-performance computing
HPSS	High Performance Storage System
I/O	input output
IDL	Interactive Data Language visualization software
INCITE	Innovative and Novel Computational Impact on Theory and Experiment
LANL	Los Alamos National Laboratory
LBNL	Lawrence Berkeley National Laboratory
LLNL	Lawrence Livermore National Laboratory
MHD	Magnetohydrodynamics
MPI	Message Passing Interface
NAG	Numerical Algorithms Group
NERSC	National Energy Research Scientific Computing Center
NetCDF	Network Common Data Format
NGF	NERSC Global Filesystem
NICS	National Institute for Computational Sciences
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
OS	operating system
PDE	Partial Differential Equation
PETSc	Portable, Extensible Toolkit for Scientific Computation
PIC	Particle In Cell
PPPL	Princeton Plasma Physics Laboratory
PSI	Plasma Science and Innovation Center
SC	DOE's Office of Science
SciDAC	Scientific Discovery through Advanced Computing
SNL	Sandia National Laboratories
V&V	Verification and Validation

Appendix D. About the Cover

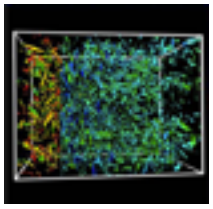


Visualization of a sparse matrix resulting from an MFDn computation showing nonzero elements in red, potentially nonzero blocks in green, and zero blocks in blue. This is after application of a new multilevel blocking algorithm. From Philip Sternberg, et al., http://unedf.org/content/annual_mtg.php.

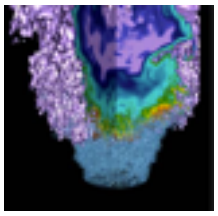


Visualization of CO_2 storage in saline aquifers showing the adaptively refined mesh. The dynamics of the density-driven velocity field induce the convective fingers that enhance the rate by which CO_2 is converted into negatively buoyant aqueous phase. Image courtesy of George Pau, Lawrence Berkeley National Laboratory.

NERSC's Cray XE6 system ("Hopper") has 153,216 compute cores, 217 TB of memory and 2 PB of disk. The system, with 6,384 compute nodes made up of 2 twelve-core AMD 'MagnyCours' processors, has a peak performance of 1.28 Petaflops/sec.



Direct numerical simulation (DNS) of a canonical shock/turbulence interaction showing turbulent eddies (worm-like objects, colored by the angular rate-of-rotation) being modified and amplified as they pass the shock. Image courtesy of Drs. Ivan Bermejo-Moreno, Johan Larsson, and Sanjiva Lele of Stanford University.



Combustion simulation of a low swirl burner. Image shows flame radical, OH (purple surface and cutaway), and volume rendering (gray) of vortical structures. Red indicates vigorous burning of lean hydrogen fuel; shows cellular burning characteristic of thermodiffusively unstable fuel. High vorticity surrounds the region of high shear generated at boundary of the annular swirling inlet. Image courtesy of John Bell, LBNL.


NERSC


U.S. DEPARTMENT OF
ENERGY
Office of Science


BERKELEY LAB
Lawrence Berkeley
National Laboratory