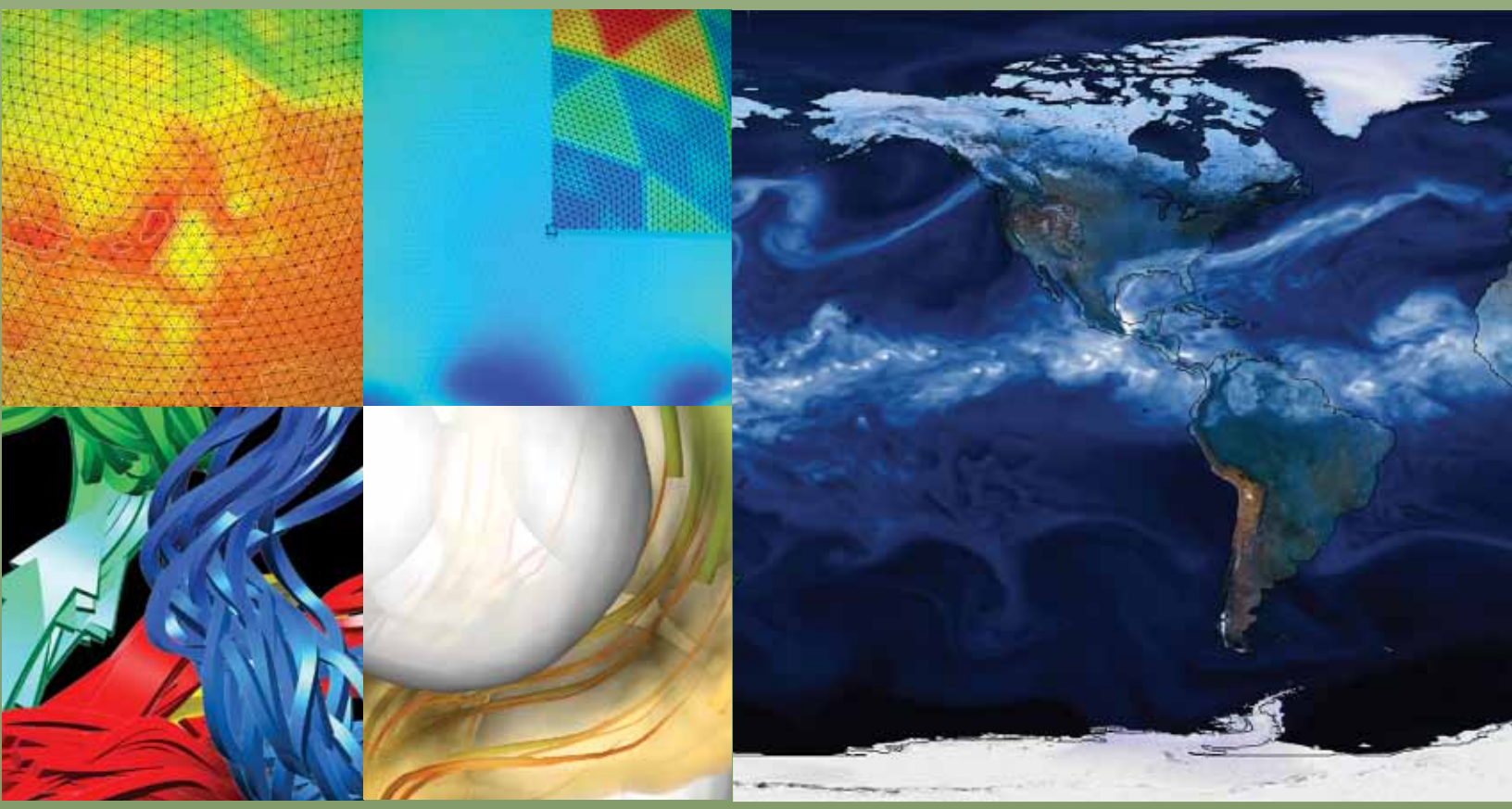
LARGE SCALE COMPUTING AND STORAGE REQUIREMENTS



Biological and Environmental Research

Report of the NERSC/BER/ASCR
Requirements Workshop
May 7 and 8, 2009

DISCLAIMER

Large Scale Computing and Storage Requirements for Biological and Environmental Research

Workshop Report

Biological and Environmental Research Program Office (BER),
DOE Office of Science
National Energy Research Scientific Computing Center (NERSC)

Washington, DC

May 7 and 8, 2009

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 Mike Riches serves as the BER 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 Directors of the Office of Science, Office of Biological & Environmental Research, and the Office of Advanced Scientific Computing Research, Facilities Division.

This is LBNL report LBNL-2710E, published October, 2009; updated October, 2010

Table of Contents

1 Executive Summary

The National Energy Research Scientific Computing Center (NERSC) is the flagship scientific computing facility for the Department of Energy's Office of Science. NERSC provides large-scale, high-performance computing (HPC) resources for scientists engaged in research that furthers the mission of the Office of Science.

To assure that it continues to meet the needs of the scientists and programs it supports, NERSC regularly gathers computing requirements from its current users, as well as from prospective user communities that are expected to embrace HPC as a tool for scientific discoveries in the near future.

In May 2009, NERSC, DOE's Office of Advanced Scientific Computing Research (ASCR), and DOE's Office of Biological and Environmental Research (BER) held a workshop to characterize HPC requirements for BER-funded research over the subsequent three to five years.

The workshop revealed several key points, in addition to achieving its goal of collecting and characterizing computing requirements. Chief among them: scientific progress in BER-funded research is limited by current allocations of computational resources. Additionally, growth in mission-critical computing – combined with new requirements for collaborative data manipulation and analysis – will demand ever increasing computing, storage, network, visualization, reliability and service richness from NERSC.

This report expands upon these key points and adds others. It also presents a number of "case studies" as significant representative samples of the needs of science teams within BER. Workshop participants were asked to codify their requirements in this "case study" format, summarizing their science goals, methods of solution, current and 3-5 year 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.

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 Biological and Environmental Research.

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 NERSC

Lawrence Berkeley National Laboratory (LBNL) operates and has stewardship responsibility for the National Energy Research Scientific Computing (NERSC) Facility, which is supported by the Office of Advanced Scientific Computing Research (ASCR), as a national resource. NERSC serves about 3,000 scientists annually and about 400 projects that use hundreds of distinct application codes. These scientists, working remotely from Department of Energy (DOE) laboratories, other Federal agencies, industry, and universities, use NERSC resources and services to further the mission of the Office of Science (SC). Computational science conducted at NERSC covers the entire range of scientific disciplines, but is focused on research that supports DOE's missions and scientific goals. The results of the scientific use of NERSC are documented in nearly 1,500 peer reviewed scientific papers per year as well as NERSC annual reports[1] and other materials. In addition to program office computational projects within the Office of Science, NERSC directly supports the Scientific Discovery through Advanced Computing (SciDAC[2]) and ASCR Leadership Computing Challenge[3] Programs as well as several international collaborations in which DOE is engaged. NERSC supports the computational needs of the entire spectrum of DOE open sciences.

NERSC serves a unique role in the DOE Office of Science computing portfolio that is distinct from the Leadership Facilities. The Energy Research Computing Allocations Process (ERCAP) is used to allocate computing time on NERSC systems for more than 400 projects to serve Office of Science programmatic activities, whereas the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) process is used to allocate computing time on the OLCF and ALCF systems for 12-24 projects selected from the international research community. Allocation of NERSC computing resources is controlled primarily by the Office of Science program managers through the ERCAP process, based on scientific merit and prioritization of projects within their respective research portfolios. Allocation of ALCF and OLCF resources are primarily controlled by the laboratories that operating the leadership systems via the INCITE process, which gates access based on scientific merit and the ability of the code to scale efficiently to use the entire leadership computing system. 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.

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

3 Workshop Background and Structure

The National Energy Research Scientific Computing Center (NERSC) is the flagship scientific computing facility for research sponsored by the U.S. Department of Energy Office of Science. NERSC, a national facility located at Lawrence Berkeley National Laboratory, is a world leader in providing resources and services that accelerate scientific discovery through computation. NERSC uniquely provides HPC resources and support for all offices and programs within the Office of Science.

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 requirements from a variety of sources; among them: the NERSC Energy Research Computing Allocations Process (ERCAP) allocation requests to DOE, workload analyses, DOE program managers, and scientists who use the facility.

In May 2009, the DOE Office of Advanced Scientific Computing Research (ASCR, which manages NERSC), the DOE Office of Biological and Environmental Research (BER), and NERSC held a workshop to gather HPC requirements for current and future science programs funded by BER. This report is the result. Findings from this workshop will serve as input to the NERSC/ASCR planning processes and will help ensure that NERSC continues to provide world-class resources and support to Office of Science-funded research projects. The format of the workshop and report was based on that used by the Energy Sciences Network (ESnet), which has conducted a series of similar successful workshops. However, the NERSC requirement space is considerably broader than that of ESnet and this document reflects such differences.

This document presents a number of consensus findings. In support of these, a number of "case study" summary reports are included as specific representative samples of the research conducted within BER. The case studies were chosen by the DOE Program Office Managers and NERSC personnel to provide broad coverage in climate, environmental, and biological research. However, BER funds many research endeavors in these fields and the case studies presented here do not necessarily represent the entirety of BER research. Each case study describes its scientific goals today and for the next 3-5 years, its computational method of solution, and a description of its current computing needs and expected future 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.

Specific findings from the workshop follow**.**

4 Findings

1. Without a significant increase in resources scientific progress in BER-funded research will be acutely limited by availability of computational resources and competition for allocations. The projects considered here collectively estimate a *seven*-fold increase in computing hours to meet their scientific goals for the next three to five years. Project scientists have also identified significant scientific benefit that would accrue given ten-fold, 100-fold, and 1000-fold increases in computing resources.

2. Many BER projects have mission-critical time constraints; examples include predictions for the next IPCC climate change report and the Joint Genome Institute's need to maintain a four-month update cycle for genome datasets. Such projects demand a computational infrastructure that includes powerful, yet highly reliable, resources. The need for guaranteed turn-around time for these key projects will require better reliability (addressed in finding #11) and resource reservation policies.

3. Codes that scale well to 10,000 cores still require significant development to perform well on a million cores. BER scientists have requested access to test-bed machines and assistance in choosing effective programming models so they can prepare to run on new architectures. A common strategy is to use OpenMP to add multithreaded execution capability to existing MPI programs.

4. Uncertainty quantification is a key multi-science BER activity that requires advances in fidelity of physics models, continued numerical algorithm development, and vast increases in spatial and temporal resolution. There is a critical need for NERSC to provide high-throughput development and analysis services in addition to being a production computing facility.

5. Ensemble runs (which consist of many simultaneous instances of the same parallel code, but with varying initial and/or boundary conditions) and long-running simulations that use smaller concurrency are important methods for scientific discovery in support of BER's mission.

6. Stability, availability and reliability are characteristics of modern HPC systems that are extremely important to BER researchers. Scientific productivity rivals machine "speeds and feeds" in importance and should be a consideration in procurements and in assessing center success (through metrics). Applications require the ability to "fail gracefully" and/or recover quickly and easily.

7. Data manipulation and analysis is itself becoming a problem that can be addressed only by large HPC systems. Simulation output will become too large to move to "home" institutions; therefore, NERSC needs to integrate robust workflow, data portal, and database technology into its computational environment and significantly increase real-time-accessible data storage capacity.

8. Several BER projects require efficient parallel I/O via MPI-IO and high-level libraries such as HDF and netCDF. Inconsistent or poorly performing I/O makes it difficult to achieve predictable throughput and accurately estimate allocation requirements.

9. Global reduction operations are a scaling bottleneck in several key codes, owing to the global nature of key algorithms; notably, those with a conjugate gradient (CG) solver. Highly efficient global reduction operations are needed.

10. Memory requirements on a per-core basis vary, but are generally 2 GB per core or less. The upper bound includes codes with legacy characteristics that if eliminated, could reduce memory footprint, while codes adapted to run on more restrictive architectures like the IBM BlueGene may have smaller requirements. Scientists want to use more memory per core, but realize that architectural trends may not favor this and they may need help reducing their code's per-core memory footprint.

11. Users underscored a critical need for highly available and reliable HPC resources. Long downtimes for HPC system upgrades and maintenance, system-wide outages due to hardware or software, and job failures due to transient node errors greatly reduce user productivity. Users would like the ability to access data or perform compilation on system front-ends (login nodes) when the system is down.

5 NERSC Response

1) The BER community requires a significant increase in computational resources to meet its research requirements. NERSC is implementing immediate power and cooling upgrades to its OSF facility and has long-term plans to ensure that future demand is not limited by facility constraints. NERSC's supercomputer acquisition process ensures selection of systems that deliver maximum useful work to application scientists relative to a given total cost of ownership.

2) To ensure high availability of its HPC systems, NERSC plans to have two major systems on the floor at all times, so that mature and stable systems are available as new ones are being deployed. This requires a major procurement roughly once every 2.5 to 3 years. NERSC is now deploying systems with high availability features such as external login nodes and filesystems to allow users to access data, submit jobs, and compile during system outages. NERSC will investigate methods to improve the ability of MPI codes to fail gracefully and/or recover quickly from failures.

3) By implementing resource reservation policies NERSC will ensure guaranteed turn-around time for key BER projects such as genome sequencing workflows that have stringent turnaround schedules. A queue reservation service is currently being evaluated and will be rolled out in the 2010 allocation year. These policies will also benefit BER project scientists who requested "computational beamlines," where guaranteed machine resources are reserved in advance for mission-critical computational experiments.

4) NERSC will take a lead role in investigating fine-grained concurrency programming models. NERSC is collaborating with the community to develop courseware and examples to help NERSC users evaluate alternative programming strategies and migrate to new, scalable programming paradigms. NERSC will also field experimental testbeds for emerging computer architectures (such as GPU and FPGA-based experimental platforms) to serve as proving grounds for novel programming models and hardware exploration,.

5) NERSC will work with the HPC community to explore frameworks for automating Uncertainty Quantification (UQ) workflows. Advanced UQ workflows depend on sophisticated coordination between the resource manager and automated data analysis systems. Since UQ typically requires many smaller runs, NERSC will continue to ensure that HPC systems it selects support a broad user workload.

6) NERSC will study using a coupled climate model as a benchmark instead of the standalone atmosphere code (as suggested in case study 6.2.1, below). NERSC has a long, successful track record of performance evaluation using full application benchmarks derived from the Office of Science workload. Benchmark selection is

constantly refined using input from users and thus, NERSC will also investigate benchmarks representing emerging BER computing workloads.

7) NERSC will explore queue policies and resource management mechanisms that support efficient ensemble runs to support key applications in BER. This will create synergy with the frameworks required to support UQ workflows, which also depend on ensemble runs.

8) NERSC will expand the integration of externally developed portal technologies into its computational environment. The current NERSC Global Filesystem expansion is being driven in part to support such portals. NERSC will investigate costs and personnel requirements to field commercial-grade (e.g. Oracle) databases to support robust data storage services.

9) NERSC is investing substantially in its storage and computing infrastructure to meet the needs of emerging data-intensive applications and to ensure consistent, high-I/O performance. NERSC has established runtime monitoring of filesystem performance on HPC systems to detect performance anomalies, and is investing in the optimization of key parallel I/O libraries for HPC systems.

10) NERSC will include benchmarks that stress the performance of global reductions and will explore methods to introduce users to alternative numerical methods.

11) The new NERSC-6 system will accommodate BER codes with larger memory footprints by including a substantial number of "fat memory" nodes and NERSC will attempt to do so in future systems as well. Future HPC systems are unavoidably trending towards lower memory capacity per core. NERSC will also perform runtime monitoring of its systems to directly measure job memory use. In addition, NERSC will ensure that reducing memory footprint is a key metric for evaluating advanced programming model alternatives.

6 Climate Science

6.1 BER Climate Science Overview

The DOE Climate Change Research program includes process research and modeling efforts to (1) improve understanding of factors affecting the Earth's radiant-energy balance; (2) predict accurately any global and regional climate change induced by increasing atmospheric concentrations of aerosols and greenhouse gases; (3) quantify sources and sinks of energy-related greenhouse gases, especially carbon dioxide; and (4) improve the scientific basis for assessing both the potential consequences of climatic changes, including the potential ecological, social, and economic implications of human-induced climatic changes caused by increases in greenhouse gases in the atmosphere and the benefits and costs of alternative response options. Research is focused on understanding the basic chemical, physical, and biological processes of the Earth's atmosphere, land, and oceans and how these processes may be affected by energy production and use, primarily the emission of carbon dioxide from fossil fuel combustion. The program is comprehensive with an emphasis on the radiation balance from the surface of the Earth to the top of the atmosphere, including the role of clouds and on improving quantitative models necessary to predict possible climate change at the global and regional levels.

The Climate Modeling Program sponsors projects that develop, test, and apply state-of-the-science coupled climate and earth system models, based on theoretical climate change science foundations. In order to enable sound decision-making on issues pertaining to future energy use and technology options, credible high-resolution climate change simulations are required at a regional scale. Research results from this program result in climate change projections for the 21st century using state-of-the-science dynamically coupled models. Understanding future variability and predictability of the climate system e.g., changes in major modes of climate variability, climate extremes, detecting and attributing the regional manifestations of climate change, remain significant challenges. Improved climate information at high spatial and temporal resolution is of immense significance to society and decision makers. To achieve such high-resolution simulations, both the accuracy and throughput need to be dramatically increased; thus the climate modeling activity takes advantage of emerging high performance computing and information technologies, e.g., DOE NERSC and Leadership-class Computing Facilities.

6.2 Climate Science Case Studies

6.2.1 Moderate and High Resolution Climate Change Simulations with CCSM

Principal Investigator: Warren Washington, National Center for Atmospheric Research
Contributors: Lawrence Buja and Jerry Meehl, National Center for Atmospheric Research

6.2.1.1 Summary and Scientific Objectives

The primary goal of this work is the completion of initial runs for the Intergovernmental Panel on Climate Change Fifth Assessment Report (IPCC AR5).

The Community Climate System Model (CCSM) version 3 was used to carry out the DOE/NSF climate change simulations for the Intergovernmental Panel on Climate Change Fourth Assessment Report (IPCC AR4) that was released in 2007. This breakthrough study, which enjoyed broad public acceptance, presented a clear picture of a planet undergoing a rapid climate transition with significant societal and environmental impacts. However, the basic question being asked by society has now changed dramatically. No longer are climate scientists being asked *if* human-induced climate change is occurring; now the question is, what is the local, time-evolving nature of climate change to which we must adapt? Our next challenge, then, is applying an emerging class of Earth System Models that include detailed physical, chemical, and biological processes as well as interactions and feedbacks in the atmosphere, oceans, and land surface, to carry out policy-relevant adaptation/mitigation scenarios. Thus, in 2009, the initial runs for the Intergovernmental Panel on Climate Change Fifth Assessment Report (IPCC AR5) will commence. This will involve running CCSM3.5 and CCSM4 at resolutions higher than ever possible before.

6.2.1.2 Methods of Solution

CCSM is a coupled climate model for simulating the earth's climate system developed through funding from a variety of sources including DOE and NSF. It is composed of four separate models, the Community Atmosphere Model (CAM), the Parallel Ocean Program (POP), the Community Sea-Ice Model (CICE), the Community Land Model (CLM), plus the CCSM Coupler. These execute concurrently as five separate binaries, exchanging information via MPI communicators. The individual components of CCSM3 use all-MPI, all-OpenMP, or a hybrid mix of the two paradigms to achieve efficient parallelism. The CCSM code is supported on Cray systems, IBM BlueGene, IBM Power, and Linux Clusters.

CAM3 is the fifth generation of the NCAR atmospheric model. The spectral Eulerian dynamical core is the default, although the code includes the option to run with semi-Lagrange dynamics or with finite-volume dynamics. CAM3 retains the Zhang and McFarlane parameterization for deep convection and includes prognostic equations for three water substances: vapor, small cloud water drops, and cloud ice particles, and diagnoses the production of rain and snow mixing ratios (and their fluxes) by assuming

the sources terms balance the sinks. The total parameterization package in CAM3 consists of a sequence of components: moist precipitation processes, clouds and radiation, a surface model, and turbulent mixing.

Ocean dynamics in POP are described by the 3-D primitive equations (momentum, continuity, hypostatic, state, and tracer transport) for a thin stratified fluid using the hydrostatic and Boussinesq approximations.

The Community Sea Ice Model (CSIM) is a dynamic-thermodynamic model that utilizes the elastic-viscous-plastic rheology dynamics and includes energy conserving thermodynamics of Bitz and Lipscomb with a subgrid-scale ice thickness distribution

The Community Land Model (CLM) is a biogeophysical process model of vegetation dynamics, plant physiology, land/surface radiation and surface/subsurface hydrology.

6.2.1.3 HPC Requirements

In the immediate future some dedicated, high processor count jobs will be required to provide high-resolution simulation experiments. A large number of low-resolution, ensemble simulation studies will also be required.

Four primary sets of runs will be carried out in 2009, three with the new half-degree atmosphere and one-degree ocean CCSM4 and the last with the older T42 (2.8 degree) resolution.

First, a 250 year present-day control will be run to establish a baseline climate plus a 1%/year increasing CO_2 experiment to determine climate sensitivity for one of the primary model configurations to be used in the IPCC AR5. Estimate is about 7.5M core hours.

The AR5 project is characterized by four future "Representative Concentration Pathways" defining increases in radiative forcing equivalent to 8.5, 6, 4.5 and 2.6W/m^2. We will run two of these RCP's at high resolution to look at the regional responses to these levels of forcings. If these are successful, we will increase the number of runs in the ensemble in the following year. Estimate is about 3.8M core hours.

Third, two near-term experiments will be run simulating the impact of a large, Pinatubo-class, tropical volcano erupted in the year 2010 for the two RCP's. This will assess the amount of climate change averted due to a natural event as seen in the 1980's and 1990's. Estimate is about 1.2M core hours.

Finally, the T42 climate dynamics simulations started in 2008 will be continued. Estimate is about 7.5M core hours.

A typical CCSM3 run might use something like 2,288 cores total. In this configuration, CAM runs on 512 MPI tasks each with 4 OpenMP threads, or 2,048 cores total. The land, ice, and coupler components run concurrently with each other but sequentially with

CAM on this same processor subset. POP runs on its own subset of processors, 240 PEs in this case. Combined, that gives 2,288 PEs, which happens to be a well-balanced CCSM configuration. There are similar distributions could use 5,844, 6,032, 6,512, and 10,272 total cores for both low- and high-resolution studies.

Typical runs are for about 20 real-time days and require nearly 2GB of DRAM per core. Data read/written consists of about 1000GB via 2-10GB checkpoint files. Checkpoint files are typically written about every three hours and if I/O takes more than about five percent of the total runtime we consider that as poor performance. A goal is to keep all of these data at NERSC. Runs must achieve no less than about five simulated years / wall-clock day to be useful.

Upcoming changes to codes/methods/approaches include use of a cubed sphere grid and higher-order spectral element methods, increased spatial resolution to resolve convection in both the atmosphere and ocean, and inclusion of more detailed physical processes (chemistry, biology, hydrodynamics).

The speed of these simulations is limited by strong scalability. Better scaling of the atmospheric model via a change in the dynamical methods will allow higher resolution simulations to be performed. We expect continued use of finite volume dynamical cores but with a change to the cubed sphere grid and the introduction of higher order spectral element methods. NERSC's systems are well suited for long-term climate simulations.

New climate science that would be afforded by more powerful computing systems has been discussed in several recent studies. The credibility of integrated earth system model predictions requires simulations that include important physical processes at their native spatial and temporal scales; e.g., 1 to 10 km for clouds and ocean eddies. Previous studies have suggested that sustained Petaflop performance is required for useful 10-km grid resolution and that predictions of societal and environmental change at 1-km resolution would require truly exascale computers. Specific gains also include those shown below.
We estimate an allocation of about 100 million hours at NERSC would be required to commence studies in these areas.

• Characterization of radiatively active atmospheric constituents, especially aerosols & clouds;
• Incorporation of chemical and biogeochemical processes in climate models;
• Interactions between changing climate and hydrological systems;
• Incorporation of knowledge from observational and modeling process studies into Earth System Models; and
• Implications of climate change for energy systems.

6.2.1.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Main science driver	IPCC AR4, CCSMv4	CCSM Grand Challenge, FV 0.1-deg, 11-km runs
Computational Hours	12 million	100 million
Parallel Concurrency	240 – 2,288	480 – 5,844+
Wall Hours per Run	20 days/simulation	
Aggregate Memory	408 GB – 3.9 TB	
Memory per Core	1.7 GB	
I/O per Run	1 TB, 2-10 GB checkpoint files	
On-Line Storage Needed		1 PB
Data Transfer		0 is goal
Archival Storage		

6.2.1.3.2 Support Services and Software

Given that simulation runs for this project will be done at multiple centers the data management problem and the integration of simulation results into the Earth System Grid for analysis and distribution will be critical elements of NERSC's support for our requirements. There is a group at Lawrence Livermore National Laboratory (LLNL) that has the responsibility for sharing these data; however, NERSC should have this core and maybe duplicate the core held at LLNL on rotating storage. Because it is doubtful that any one site can hold the entire database (~1PB) it would require a ~100-GB/s network between sites.

There are also unsolved software issues associated with viewing these data such as the problem of doing a high-volume "diff."

A known trajectory for NERSC system architecture, and one that tracks that of other sites, would aid our project.

A key issue is the need for stable and reliable platforms for this work over the next three years to support the goals of the assessment work and ongoing development of scalable and extensible earth system models. This goes beyond just statistics regarding job completion, available cycles, etc.. It goes to user *perception* of system stability. A climate researcher has to choose a center for high-priority runs and the one *perceived* as being most stable might well be the choice.

It also helps to have architectural balance between compute, storage, and data transfer speeds. We view NERSC as both a development and a production center so it is vital that NERSC maintain capabilities for both. We prefer a batch environment for development

so we can duplicate the environment of real runs and so we can document what happened.

We like the idea of NERSC doing benchmarking and evaluation using NCAR's climate codes; however, NERSC should consider using the entire coupled simulation instead of stand-alone versions of the components.

We would like to explore incorporating modern workflows into our process but are not sure how to do this.

6.2.1.4 Emerging HPC Architectures and Programming Models

CCSM's hybrid OpenMP/MPI design allows each component model to find a "sweet spot" in current HPC architectures and in particular, to exploit the multi-core expansion of the Cray XT4 and XT5. We would like to investigate the advantage of specialized processors or additional structure in the memory hierarchy but we see this as a challenge blocked by lack of portable programming standards and techniques.

6.2.2 Coupled High-Resolution Climate Modeling of the Earth System

Principal Investigator: V. Balaji, Geophysical Fluid Dynamics Laboratory (GFDL)
Contributors: Christopher Kerr, GFDL

6.2.2.1 Summary and Scientific Objectives

The goal of this joint NOAA/DOE work is to give an early look at issues associated with resolving mesoscale features in atmospheric and ocean circulations, with implications for understanding forced and natural variability of the climate system. Although high-resolution models can provide improved understanding of regional climate change, current approaches to generating consensus and uncertainty estimates rely on statistical methods comparing results from many models. The design of model comparison studies is based in part on the understanding of the behavior of a known suite of models at some target resolution. Results from our simulations will provide near-term insight into regional climate change and may inform the design of international modeling campaigns aimed at addressing those questions. These are key scientific issues that centers will have to tackle independently before making the leap to higher resolutions. The science therefore, supports the NOAA and DOE missions of providing a predictive understanding of climate change and credible and timely information for decision makers in preparing for climate change.

6.2.2.2 Methods of Solution

The GFDL weather and climate models are built on the Flexible Modeling System (FMS), a framework that enables different components of the climate system (ocean, ice, atmosphere, land) to be constructed by independent groups of scientists, algorithm developers and software engineers and assembled in a variety of ways. Key goals in FMS software design were portability and efficiency on a wide range of distributed and shared memory platforms. The FMS infrastructure is written in Fortran 90 with some C language modules and includes software to handle parallelization, input and output, data exchange between various model grids, orchestration of time stepping, makefiles, and simple sample run scripts. Underpinning this framework is a highly scalable parallel communication fabric that can exploit shared (OpenMP) and distributed memory (MPI) approaches in a manner that is transparent to "user code" the science modules within the model. The I/O scheme used is critical to the overall performance. The I/O is multi-threaded where output is written to independent files that are combined at post processing. All files are read and written in NetCDF. FMS also includes software for standardizing, coordinating, and improving diagnostic calculations of FMS-based models, and common preprocessing and post-processing functionality not adequately provided by available third-party software.

A cubed-sphere atmosphere dynamical core is now a default at the GFDL to eliminate some of the scalability issues associated with the finite volume core on rectangular grids. The ocean model in these experiments is the Modular Ocean Model (MOM), which uses

a tripolar grid and also uses subcycling instead of an elliptic solver to eliminate the CG solver scalability issue that arises in POP.

6.2.2.3 HPC Requirements

The models typically run on distributed memory platforms and have extensive memory requirements at 2 GB per MPI task, although much of this may be due to poor programming practice and could potentially be reduced by as much as a factor of four.

Computational experiments will require approximately 5TB of "scratch" space per user. The intention is to transfer all data produced to the GFDL for post processing and analysis, for several reasons. First is complexity – the analysis software involves using postscript files where a user selects some subset of data; currently four people at GFDL are dedicated to supporting this software infrastructure and there is no plan to port it. Second is size – the data sets "explode" when doing the post-processing, sometimes three- to seven-times the original size. Third is that the GFDL considers the data to be proprietary. The GFDL has recently upgraded to the Energy Sciences Network. The intention is to generate data at a rate of 1.6TB per wall-clock day and the current networking plans should satisfy this requirement.

The GFDL is able to run some smaller, low-resolution studies on an internal SGI Altix system but collaboration with the DOE is required for the higher-resolution runs. Several standalone atmospheric model experiments are running in production now at NERSC and the Oak Ridge Leadership Computational Facility. These include:

- ½-degree (25-km) runs with several scenarios for a total of 100 model years. The computational performance is about 4 model years per wall-clock day on 1,400 cores. These runs produce about 100 GB/model-year.

- 1/4-degree (50-km) runs consisting of several scenarios running for a total of 300 simulated years. Computational performance is about 2 model-years per wall-clock day on 1,944 cores and about 300 GB/model-year are produced.

Several experiments are in development with plans to run soon on DOE systems, including the Argonne Leadership Computational Facility. These include:

- ⅛-degree (13-km) standalone atmospheric model experiment planned for 10,000 cores;
- Coupled runs using ½-degree (50-km) atmosphere and ¼-degree (25-km) ocean models in which several scenarios will run for a total of 200 years. The computational performance is expected to be about one model-year per wall-clock day on 2400 processors, producing about 70 GB per year.
- Coupled runs using ½-degree (50-km) atmosphere and 1/10-degree (10-km) ocean models for 200 model-years, possibly on ~20,000 cores.

The experiments with the ¼-degree standalone atmospheric model and the coupled climate model are intended for submission to the Intergovernmental Panel on Climate

Change Fifth Assessment Report. Because of the memory limitation on ALCF IBM BG/P ("Blue Gene") machine and because our runs usually require about two GB per MPI task, development runs on BG/P are using only one MPI task per node.

A rule of thumb for "scientifically useful" runs is a minimum of three model years / wall-clock day (100 model years / wall-clock month). Current projected throughput is marginal and assumes dedicated CPUs (no queue wait time).

There is a need to do decadal (and longer) predictability studies and explore hurricane statistics under climate change. Both require ensemble runs that would easily use as much allocation as is made available given other constraints such as data volume.[4]

6.2.2.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	2.5 Million (1.05 at NERSC)	150 Million at NERSC, ALCF, & OLCF
Parallel Concurrency	864 – 2,400	20,000 – 80,000
Wall Hours per Run	4	
Aggregate Memory	1.7 – 4.8 TB	10 TB – 160 TB
Memory per Core	2	<2 with MPI/OpenMP ?
I/O per Run	4.5 – 90 TB est.	14 TB est.
On-Line Storage Needed		
Data Transfer		

6.2.2.3.2 Support Services and Software

One of the principle challenges is the ability to debug and profile the models at high processor counts. We would like to have the "debug" queue's characteristics scale appropriately with size of system; it would be even better to have a separate, somewhat smaller machine of the same architecture for debugging.

Once the models have been running they seem to require constant monitoring. Approximately 15 percent of the experiments crash because of hardware, OS, and file system problems (climate models are extremely IO intensive). High availability – something like 95 percent – is required for AR5, which places a time limit on generation of our results. We would also like to see improved job turnaround. Our jobs sit in queue for long time, run for the maximum queue length, are then (re)queued, and this cycle is repeated.

[4] V. Balaji, "Coupled High-Resolution Modeling of the Earth System," Fall Creek 2008.

6.2.2.4 Emerging HPC Architectures and Programming Models

Key tasks for GFDL going forward include:

- Minimizing the memory footprint of the model infrastructure;
- Implementing OpenMP into the model components;
- Developing a coupler that extends to 100,000 processors; and
- Optimization of parallel I/O.

The current software infrastructure has been built for O(100-1000) processing elements and is scalable to O(10,000). However, our work has shown that significant development is still needed to extend the scalability to the 100,000-million processor range. The ability to utilize this processing power will have a significant impact on the ability of the GFDL to perform high-resolution climate experiments for the next generation of models used in the Intergovernmental Panel on Climate Change.

The entire FMS modeling infrastructure currently uses MPI and we are in the process of moving to a hybrid model and incorporating OpenMP into the various component modules such as atmospheric dynamical core, physics, chemistry, ocean, and land. We have OpenMP working in some of these component modules (such as atmosphere dynamical core and physics) and will be working on the other components next. The OpenMP implementation in the modules we have completed "hides" the implementation from the scientist and this work on the component modules has helped define the programming structure. The severe memory limitation on the BG/P system is an important reason why we have been moving aggressively towards implementing the hybrid (MPI-OpenMP) mode into the entire model infrastructure. The atmosphere dynamical core does run in hybrid mode on BG/P and we are currently testing the implementation. The rest of the infrastructure should be hybrid enabled in the next 6-months to enable it to run on BGP. On the Cray XT5 we are expecting OpenMP parallelism to enable the move from about 2,000 cores to about 20,000. Early experiments suggest 7.5/8 speedup on the XT5.

The GFDL has chosen to continue development of its software infrastructure on the DOE platforms at ANL, NERSC, and ORNL. Both guaranteed access to these platforms and the software expertise at the DOE Laboratories are critical to the success of our understanding of climate and weather systems.

6.2.3 Climate and Weather Simulations with a Global Cloud Resolving Model

Principal Investigator: David Randall, Colorado State University
Contributors: Ross Heikes and Celal Konor, Colorado State University; Marat Khairoutdinov, State University of New York at Stony Brook

6.2.3.1 Summary and Scientific Objectives

The overarching goal of this work is to do kilometer-scale global climate modeling in which we resolve important processes that are currently parameterized. Principal among these are the deep convective processes responsible for the transport of moisture from near the surface to higher altitudes. The key to accomplishing this is development and testing of a global cloud-resolving model (GCRM). Creating a GCRM is important because physical parameterizations used to represent clouds in coarser models are still problematic and are likely to improve only gradually over the coming years. Fluid motions of the atmosphere exhibit no convenient "spectral gaps" that motivate any particular grid spacing and while the finest feasible grid is used to resolve as many of the energy-containing scales as possible, as grids are refined, the equations themselves must change in order to represent the dominance of different physical processes on different scales. GCRMs are likely to be used for operational numerical weather prediction within about ten years. Shortly thereafter, GCRMs will be used to perform "time slice" simulations within longer climate change simulations using coarser grids. Although they are just barely feasible on today's machines our goal is an annual cycle simulation by the end of 2011.

6.2.3.2 Methods of Solution

Our code is a global coupled atmosphere-ocean-land surface model that uses finite-volume methods to solve the Reynolds-averaged Navier-Stokes equations of motion on a sphere using a turbulence parameterization. We use a non-hydrostatic dynamical core and a geodesic grid derived from the icosahedron. Vertically propagating sound waves are filtered in the continuous equations, without the use of a reference state. Because of the dominance of vorticity in meteorologically relevant motions on all scales, the model integrates the 3-D vorticity equation directly instead of using a 3-D momentum equation. The geodesic grid and a new explicit horizontal differencing scheme permit relatively large (10-second) time steps. A very efficient, two-dimensional, parallel multigrid elliptic solver, for which the computational scaling is linear in the number of grid cells, is used. The model includes height coordinates for now; we plan to experiment with a quasi-Lagrangian vertical coordinate in the future.

Our simulation method lends itself well to parallelization, which is currently done using pure MPI. Successful simulations with moderately high (but not cloud-resolving) resolution using about 600,000 grid columns have been done as have tests on Franklin (NERSC's Cray XT4) and Jaguar (Oak Ridge National Laboratory's Cray XT5) with 30-160 million grid columns and good results for both strong and weak scaling. Some observed run-to-run performance variability is of concern. The multigrid solver exhibits

superlinear speedup in testing on Jaguar with up to 180,000 cores and typically takes no more than about 15 percent of the runtime.

6.2.3.3 HPC Requirements

Our immediate goal is a global atmospheric model with a uniform global horizontal grid spacing of 4 km or better ("cloud permitting"), 100 or more layers, parameterizations of microphysics, turbulence (including small clouds), and radiation, and an execution speed of at least several simulated days per wall-clock day.

A typical model run would include ten 3-D prognostic and ten 3-D diagnostic fields, which, at 40 million grid columns and 100 layers, equates to about 400 GB of data for a full write. In tests on Franklin with 20,480 cores and a 4-km grid spacing (40 million grid columns), a full dynamical core can produce about ten simulated days per wall-clock day. Performance expectations are that the full model with physical parameterizations using the same grid will produce about five simulated days per wall-clock day on 20,000 cores. This would consume about 50,000 processor hours/simulated day with 4-km grid spacing or about 20 million processor hours/simulated year.

Thus, this project requires substantially larger allocations of computer time to make headway towards achieving our true science goal. Specifically, we would like to complete two annual cycles with a coupled ocean-atmosphere model and 4-km resolution; we estimate this would require about 40 million hours (Franklin equivalent), or about 20 times our 2008 NERSC allocation. Looking forward to 2015 we would like to do a ten-year, 4-km simulation and also several month-long test runs with 2-km resolution but this would consume about 100 million hours. Using the same scaling logic a full, ten-year 2-km simulation would require about 1.5 *billion* hours.

Our longer-term target resolution is a horizontal grid spacing of about 1 km. This corresponds to approximately 671 million grid columns, each with about 100 layers. We believe that achieving this goal at a practical computation rate will not be possible using conventional architectures; thus, it is the target resolution for the Green Flash project ("*A New Breed of Supercomputers for Improving Global Climate Predictions*": http://www.lbl.gov/cs/html/greenflash.html).

Restart files are expected to be relatively insignificant but diagnostic or "history" files, which provide detailed information on the progress of the simulation, will be substantial. The number of diagnostic fields is controlled at runtime and the volume of a single record will approach 16 GB (per gridpoint) for a 3-D field at 4-km resolution and 256 GB at 1-km resolution with 100 vertical levels. In many cases, these files would be written once per simulated-month, but depending on the nature of the study and capacity of the IO and storage system they might be written as often as once per simulated day or even hourly or three hourly.

The total output volume produced could therefore, be as large as several petabytes. Strategies to catalog, browse, subsample, and transport the output rapidly and efficiently need to be developed. Data will need to be staged from disk to archival media during the

run since available on-line storage is currently limited to the 10s of TB range at centers such as NERSC. While it is obviously desirable to avoid transporting huge volumes of data around the country via networks or otherwise, some long-distance data transportation, containing subsamples of the GCRM output, will be necessary and for larger data volumes is impractical at today's rates. Some of this service has been built into a prototype data portal that is currently installed at NERSC. The long-term vision is to move the features specific to GCRM into the Earth System Grid portal software rather than duplicating that extensive effort. Still, major portions of our data analysis and visualization work will have to be carried out at the same center where we perform the simulation.

2011: Two annual cycles with a coupled ocean-atmosphere model (Grid 11) @ 40 million hours (Franklin equivalent). This is expected to double yearly out to 2014.

2015: Ten-year simulation @ 200 million hours (Grid 11) and also several month-long test runs with Grid 12 @ 100 million hours

2018: Ten-year simulation @ 1.5 billion hours (Grid 12)

6.2.3.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Main science driver	Lower-resolution, cloud-parameterized	1-2-km, cloud resolving
Computational Hours	1 million	80-320 million
Parallel Concurrency	20,000	80,000+
Wall Hours per Run		
Aggregate Memory		
Memory per Core		
I/O per Run	~0.4 TB	1PB+
On-Line Storage Needed		
Data Transfer		
Archival Storage	50 TB	<=1PB

6.2.3.3.2 Support Services and Software

The main computational challenges faced by our project are:

- Efficient execution on a very large number of processors, to achieve acceptably fast run times;
- Parallel I/O (especially O; see below);
- Management and distribution of the voluminous model output; and

• Analysis and visualization.

Regarding visualization, we have enjoyed considerable support from the NERSC visualization group in initial development of new visualization infrastructure for icosahedral grids. However, our target model and grid resolution will be greater than devices can handle and what the eye can see. We need a way to roam through the data produced by ensemble runs, something like GoogleEarth to do zooming or analysis on data within the current field of view. These tools don't exist and they need to be community based.

Regarding I/O, the climate community has historically used NetCDF because a latitude/longitude formalism is implicitly built in to it; however, it may not work as well for icosahedral (or quadrilateral) grids. So we require support for netCDF but this may not be sufficient for our needs.

A separate partnership has been developing an I/O API for our code that linearizes data from the icosahedral grid and writes blocks of data contiguously in a single file in processor-independent order. The API is designed to support multiple parallel (or serial) higher-level I/O layers, including pNETcdf, NETcdf4, and NETcdf3 but the key point is that we require an efficient MPI-IO layer. We have also drawn upon expertise and significant resources of NERSC staff and Cray engineers to make progress in this area but if the model is to migrate to other architectures with different performance characteristics we might encounter similar difficulties with parallel IO and require similar center support. We also plan to experiment with HDF5.

We also use IDL, Mathematica, and Visit.

6.2.3.4 Emerging HPC Architectures and Programming Models

There are fundamentally two issues associated with atmospheric modeling on highly parallel machines. The first is the level of parallelism available and in this regard the GCRM using an icosahedral grid is well positioned to take advantage of systems with millions of processors because there is ample parallelism using multi-million grid columns and the data structure is easily decomposed across processors. The second is the manner in which the parallelism is expressed. In this regard, the task of recoding to use the latest machines seems qualitatively more challenging than it has ever been. Currently, our GCRM code uses MPI-1 only. OpenMP has not yet been implemented but would allow some speed-up and we intend to try it. Our method of parallel decomposition ensures some locality of subdomains that would lend itself well to an increasing number of cores per socket. Parallelization within grid columns is another possibility. Future plans also include use of remote memory addressing via MPI-2 communication (MPI_PUT or MPI_GET) to extend the MPI-everywhere model and also Partitioned Global Address Space (PGAS) languages such as Berkeley UPC.

6.2.4 The Role of Eddies in the Meridional Overturning Circulation (Ocean Modeling)

Principal Investigator: Paola Cessi, Scripps Institute of Oceanography
Contributors: Christopher Wolfe, Scripps Institute of Oceanography

6.2.4.1 Summary and Scientific Objectives

Our project seeks to understand the deep ocean circulation and its response to an altered atmospheric composition. It is believed that the main oceanic thermocline (which separates the cold deep ocean from the warmer surface layers) and heat transport to high latitudes is mediated by ubiquitous mesoscale oceanic flows driven by surface winds and differences in solar heating. We study the fundamental dynamics of these flows using high-resolution models of the ocean over moderate scales and a wide range of external parameters such as wind speed and surface temperature. Because of their relatively small scale, general circulation climate models do not resolve the mesoscale flows, and their statistics are poorly known from observations. However, we believe that they are an essential component of the ocean-atmosphere heat budget and a major player in sequestering CO_2 into the deep ocean.

6.2.4.2 Methods of Solution

The main tool for carrying out our simulations is the Massachusetts Institute of Technology general circulation model (MITgcm), which time-steps the three-dimensional Navier-Stokes equations with rotation and gravity in a variety of geometries. We use the primitive equation version of this model, (i.e. the hydrostatic, incompressible and locally constant density, i.e., Boussinesq approximation). The pressure diagnosis requires solving an implicit two-dimensional (2-D) Poisson problem.

Current simulations solve the hydrostatic primitive equations via horizontal domain decomposition; all communication is handled by MPI. Most of the computation is tile-local, but each time-step requires the solution of a global 2D elliptic problem that is solved using a preconditioned conjugate gradient algorithm. The global nature of this algorithm creates the strongest constraints on the efficiency and scalability of the code. We find that we can spend as much as 1/3 of the computational time waiting for global reductions required by the conjugate gradient algorithm to complete.

The hydrostatic approximations provide a valuable and computationally tractable model, but may not faithfully capture a number of important physical effects. For one, they cannot properly simulate convection. Most of the volume of water in the ocean is ventilated at high latitudes via convection, so if convection is not done correctly the properties of the abyssal water masses might be set incorrectly leading to an inaccurate picture of deep stratification and the overturning circulation. The convection scheme we are using is a standard parameterization, but there are convincing arguments that correctly resolving convection can lead to a qualitative change in the structure of the overturning circulation. Running a non-hydrostatic global ocean climate model would be quite an undertaking, so this is more of a long-term goal.

A second circumstance where non-hydrostatic effects become important to ocean climate is near the ocean boundaries. Due to vorticity dissipation constraints, most of the water downwelled in the downward branch of the global overturning circulation does so near the boundaries. Here the aspect ratio of the circulation approaches one and non-hydrostatic effects become important. We have an analytic model of these downwelling boundary layers, but we will need a non-hydrostatic model to test the analytic model. These tests will be done in the near term since we will take a regional modeling approach requiring much less computational effort than a global non-hydrostatic simulation.

6.2.4.3 HPC Requirements

Our current production configuration uses 1024 cores on Franklin (2048 cores on Intrepid at ALCF) and advances the simulation at a rate of approximately two simulated years per wallclock-hour. Each run requires 100-500 years to equilibrate, thus requiring 50k-250k core-hours or 50-250 wallclock-hours per simulation. Each job spends, on average, three times as much time waiting in the queue or waiting for the machine to come online as it does running; thus, the total time to complete a run ranges from 6 days to 1 month to complete. This is an acceptable throughput for our current needs. Our memory and storage requirements are modest (< 0.25 GB per core, 100 GB online storage). Our calculations are not significantly memory or storage limited.

Memory bandwidth and all-reduce performance are bottlenecks for our code. On Franklin, approximately one-third of the computational time is spent performing global reductions as part of the pressure solver. On machines optimized for collective operations (such as the IBM BlueGene series), the overhead of the global reductions is much reduced.

Anticipated future studies will focus on regional dynamics and require non-hydrostatic simulations. These will require the solution of a 3D elliptic problem at each time step and are expected to substantially increase the computational burden of our research. Our expected needs for the three- and five-year time horizons are as follows:

- In the next 1-3 years, we expect to do basin-scale hydrostatic simulations at increased resolution and non-hydrostatic simulations in small domains. These will require between 1k-5k cores per job, but only modest increases to the amount of memory per core and no more than a factor of five increases in availability of storage. In order to maintain a throughput of at last 5 simulated years per wallclock-day, we will likely have to make modest modifications to the code, primarily involving improvements to the elliptic solver. A machine with an efficient global-reduction network (like the BlueGene) would allow us to scale to a larger number of cores more easily, but we are unlikely to be seriously constrained by the particulars of the next generation machines.
- Ultimately, we hope to be able to do basin-scale non-hydrostatic simulations; these sorts of calculations are currently impossible on a large-scale, but we hope to be able to undertake them within the next 3-5 years. Such a calculation would require more than 100k cores. As with our current simulations, we expect the solution of the elliptic problem to be a major constraint on scalability and

efficiency; careful attention to the hierarchical nature of multi-core machines will be required if we are to make any headway.

The large-scale non-hydrostatic simulations are not expected to require significantly more memory per core, but will place significant demands on storage systems: each checkpoint file would consume 15 TB of disk space and the simulation would need access to 1 PB of online storage

6.2.4.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	7 Million	5-10 Million
Parallel Concurrency	1,024	Up to 100,000
Wall Hours per Run	250	1,000
Aggregate Memory	250 GB	250 GB – 1 TB
Memory per Core	0.25 GB	0.25 GB
I/O per Run	6 GB	1 PB
On-Line Storage Needed	100 GB / 5,000 Files	1 PB / 5,000 Files
Data Transfer	2 GB / day	
Archival Storage	100 GB / 400 Files	300 GB / 500 Files

Our experience with the current generation of supercomputers leads us to value reliability and simplicity over pure computing power, and we expect that any MPI-based platform that satisfies the requirements of the larger climate modeling community will be adequate for our project. Future machines should be more hardware fault-tolerant than the current system (i.e. Franklin). Our major requirements going forward are increased machine stability and tools to simplify development on the highly heterogeneous platforms likely to be in place in the near future.

6.2.4.3.2 Support Services and Software

We would appreciate a focus on improving the reliability of future systems and simplifying the programming models that will run on them.

Moving the large amounts of data that will be produced from large, non-hydrostatic simulations to our local workstations for analysis would be impractical and highlights the need for advanced analysis capabilities to be hosted at NERSC.

6.2.4.4 Emerging HPC Architectures and Programming Models

It is not clear that a viable programming model currently exists for systems with large numbers of cores-per-CPU and NERSC would do a great service by helping to provide one. The MITgcm code was originally developed as a mixed OpenMP/MPI code, but the current code is not thread-safe so the code is effectively MPI-only. The code will have to be made thread-safe and run in a mixed MPI/OpenMP mode to properly take advantage of multi-core CPUs. The general infrastructure for maintaining a distinction between "local" tiles and "remote" tiles exists in the code, we just need to make sure it works properly.

6.2.5 Atmospheric Boundary Layer Studies

Impact of Vegetation on Turbulence Over Complex Terrain: a Wind Energy Perspective

Principal Investigator: Edward Patton, National Center for Atmospheric Research (NCAR)
Contributors: Peter Sullivan, NCAR

Influences of the Boundary Layer Flow on Vegetation-Air Exchanges of Energy, Water and Carbon Dioxide

Principal Investigator: Xuhui Lee, Yale University
Contributors: Edward Patton, NCAR

6.2.5.1 Summary and Scientific Objectives

The objective of our work is to establish a mechanistic understanding of turbulence and turbulent transport in the atmospheric boundary layer (ABL) and the impact that vegetation and land-surface heterogeneity have on the exchange of momentum, energy and trace gases between the land surface and the overlying free atmosphere. Of critical importance are: 1) the land-surface and the vegetation covering it tend to act as critical sources or sinks of these entities, and 2) ABL turbulence exhibits extremely complex responses to forcing imposed across widely disparate space-(spanning from a millimeter to hundreds of kilometers) and time-scales (ranging from milliseconds to days). Weather and climate models are typically unable to resolve the ABL. Therefore, in order to accurately represent the exchange between the land-surface and atmospheric layers outside the ABL, these models require simple parameterizations to account for the ABL's diurnally varying response to land-surface characteristics and variability. A mechanistic understanding of ABL turbulence is also essential for improved wind turbine design, turbine deployment strategies, wind resource assessment, and interpretation of in-situ observations. Turbulence-resolving computations of the ABL provide the ability to isolate and identify critical dynamical ABL processes, linkages, and responses to parameter variations leading to the development of more accurate ABL parameterizations for weather and climate models that transition naturally between forcing regimes and to enhance efficiency and cost effectiveness of wind energy capture.

6.2.5.2 Methods of Solution

We use NCAR's large-eddy simulation (LES) code that predicts time dependent velocity and scalar fields by integrating the Navier-Stokes equations and heat conservation equations on a Cartesian grid. The various flavors of this code are specifically designed to investigate geophysical turbulence and its coupling with the Earth's complex interfaces at widely disparate scales of interaction. The basic algorithm is mixed pseudo-spectral (FFT) finite-difference with third order Runge-Kutta time stepping. Land-surface boundary conditions vary with the problem and

are sometimes fixed or specified via input but we also couple the LES code to the NOAH land model (the primary land-surface model in NCAR's Weather Research and Forecast [WRF] model) to allow time-dependent, spatially varying surface boundary conditions based upon predicted soil moisture/temperatures, canopy photosynthesis, and atmospheric demand.

The NCAR-LES is written in FORTRAN 90, relies heavily on serial 1-D real and complex FFT routines in NCAR's FFTPACK library, and uses a 2-D parallel decomposition via MPI. Communication involves a combination of ghost point exchanges and custom MPI matrix transposes that require only local communication; i.e., communication between processes in groups, not global MPI ALL_to_ALL. These transpose routines are also used when solving the elliptic Poisson equation for pressure.

The version of the NCAR-LES that utilizes 2-D decomposition does not currently permit complex topography. For studying orographic/vegetation/atmosphere coupling, we use an older version of the NCAR-LES that uses a curvilinear coordinate system and a 1-D parallel decomposition in which topography can be only 2-D. We are actively developing a version with a 2-D decomposition for 3-D terrain that can be time-fixed (turbulent flow over 3-D hills) or time varying (mimicking turbulent flow over 3-D water waves). This code is about two times slower than our flat code because we need to iterate for pressure.

During a run there are three types of I/O: 1) history files containing the time evolution of averaged quantities written as direct-access files solely by the "root" process; 2) 2-D time series planes of instantaneous quantities written at a relatively high frequency (~ 100 time steps) as single IEEE binary files written in a round-robin fashion using MPI I/O; and 3) complete 3-D checkpoint files read/written as single IEEE binary files using MPI I/O. At a minimum, these 3-D checkpoint files contain five 8-byte real variables (three velocity components, the subfilter-scale energy, and temperature) at every grid point, but could increase to nearly 60 8-byte variables when we incorporate NCAR's MOZART chemical mechanism within the NCAR-LES to study water vapor, CO_2, and other trace gases.

6.2.5.3 HPC Requirements

Problems at NERSC have not yet involved complex chemistry and have ranged from 256^3 up to 1024^3 grid points running on 64 to 4,096 cores. Timing tests suggest good scalability for both strong and weak scaling over a wide range of problem sizes. Sample calculations on Franklin using $1{,}024^3$ grid points with a 5,120 x 5,120 x 2,048 m^3 domain at 5 x 5 x 2 m^3 resolution running on 4,096 cores takes about 60 wallclock hours per two hours simulated time (250K Franklin core hours). Similar calculations on $2{,}048^3$ grid points running on 8,192 CPUs require about 1.8M Franklin core hours per hour of calculation.

Calculations are typically limited by time, meaning that we need to integrate for long durations, since the CFL condition (maximum time step needed to faithfully model the dynamics) requires a 3-second time step. Simulations are usually integrated forward for 4-20 hours of simulated time just to allow turbulence to develop and come to equilibrium with the forcing; at this point the analyses begin. Depending on the case, stable statistics require about five large-eddy turnover times for averaging, which usually requires 2-10 hours of additional simulation.

Checkpoint frequency depends on problem size. For smaller calculations (256^3 – 512^3 grid points), checkpoint files are written every one to 5,000 time steps. For larger problems ($2{,}048^3$) checkpoint files become quite large (0.5TB). Due to the queue design on Franklin (which biases jobs requesting large numbers of CPUs), size limitations on the /scratch file system (with special dispensation our limits were increased to 1.5TB), and the overhead associated with writing this much data, our operating strategy has been to request large numbers of cores for smaller duration and only write a single checkpoint file at the run's completion.

When we output 2-D planes of instantaneous quantities each slice-type spans the 'bricks' differently; therefore, unless we perform a transpose, one of these planes requires every MPI task to write a very small data record to the file, which exhibits low performance.

As our problems get larger, we find that we prefer not to output very many 3-D volumes because they are extremely large and the I/O is time-consuming. For this strategy to be successful, the machines we're running on need to be reliable enough that we can count on the computation arriving at the designated time for checkpointing so that we can write a restart file.

Petascale computing has the potential to alter the landscape of turbulence simulations in ABL computation and allow stratified turbulent flow simulations over a wider range of scales (meters to tens of kilometers) and in more realistic environments such as undulating terrain in the presence of vegetation. This will allow us to resolve 1-10 meter surface features while still capturing 1-100 km energy scales of motion in the boundary layer. These kinds of calculations have been impossible on existing computational platforms.

A key goal is studying turbulence driven by time-varying forcing (e.g., including the effect of sunrise/sunset) but this will require averaging of ensemble runs - up to 100 - each for a full diurnal cycle. To pursue these multiple, much longer-running calculations, we will use a newly designed, massively parallel, boundary layer code with general topography and a coupled multi-level canopy land-surface model.

Unconstrained over the next five years, we would run the equivalent of about 100 of the previously described $1{,}024^3$ simulations (25M Franklin hours) with parameter space variations spanning a wide range of topographic complexity, vegetation density/type, spatial heterogeneity, solar forcing, and temperature/water/carbon

scenarios. These runs would allow considerable additional model fidelity: 3-D hills, time-dependent grids to mimic 3-D water waves, clouds, chemistry (which would add 50+ more scalars), and/or coupling of canopy source/sinks (couples trees and atmospheric demand for moisture). All of this will be vital for capturing vegetation influence in global climate models

6.2.5.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Main science driver	Portion-of-day simulations, fixed forcing	More general and time-varying topography, time-varying forcing, full-diurnal cycle.
Computational Hours	500,000	1 – 10 million
Parallel Concurrency	64 - 4,096	1,024 – 16,384
Wall Hours per Run	12-600 hours	
Aggregate Memory	128 GB – 8 TB	
Memory per Core	2 GB	
I/O per Run	2 GB – 2 TB	
On-Line Storage Needed		
Data Transfer		
Archival Storage		

6.2.5.3.2 Support Services and Software

Our needs include methods for simulations to recover from node or I/O failure, and guidance toward efficient usage of forthcoming hardware/software infrastructure (multi-core).

6.2.6 The Role of Climate System Noise in Climate Simulations

Principal Investigator: James Kinter, Center for Ocean-Land-Atmosphere Studies (COLA)
Contributors: Cristian Stan, COLA; Ben Kirtman, University of Miami; Cecilia Bitz, University of Washington; John Dennis, Richard Loft, and Mariana Vertenstein, NCAR

6.2.6.1 Summary and Scientific Objectives

Simulations supporting the scientific consensus that human activity is changing the Earth's climate have been derived from models run at coarse, O(100 km), resolutions. The impact of unresolved scales on these predictions is not precisely known. Indeed, it has been hypothesized that noise in the climate system (fluctuations on short spatial and temporal scales) could be "reddened," thereby influencing the low-frequency components of the climate signal. If true, incorrect simulation of the noise statistics (or stochastic forcing) due to inadequate resolution or errors in the physical parameterizations could feed back onto the mean climate. If this happens, the impact on future climate simulations could be enormous. It would mean that modeling improvements, such as better physical parameterization of unresolved scales, perhaps combined with higher resolution, would be necessary to model climate variability correctly. That conclusion could increase the computational cost of future climate studies by many orders of magnitude. If the hypothesis is proven false, i.e., if increased resolution does not change climate variability significantly, then we can proceed with much of the current low-resolution research program intact. To shed light on these issues this project seeks to run high-resolution, century-long simulations of the Earth System that are designed to test the importance of noise at unresolved scales.

6.2.6.2 Methods of Solution

The main tool for carrying out our simulations is the Community Climate System Model (CCSM). Our particular configuration of CCSM couples a 0.5-atmosphere and land model to a 0.1-ocean and sea ice model. We are also using an interactive-ensemble method to explore the role of noise. Instead of using just a single atmosphere or ocean model, our configuration may include 10 atmosphere or ocean models whose state is averaged. CCSM has the following component models: The Parallel Ocean Program (POP) will use 3600x2400x42 grid points., the Community Atmospheric Model (CAM) uses a finite volume method with 576x384x30 grid points, the Community Land Model (CLM) uses 576x384x17 grid points, and the Community Ice CoDE (CICE) uses 3600x2400x20 grid points. The component models are coupled together using the CPL7 coupler.

Our current production code performs all disk I/O through a single MPI task per component model. Development versions of two of the five components have parallel I/O. We expect all components to support parallel I/O before the end of 2009.

6.2.6.3 HPC Requirements

Computing support for our project is currently provided through a TeraGrid grant of 35M hours at NICS, a 2.3M-hour "Directors Grant" at NERSC from late 2008, and a Grand Challenge grant of 7.7M hours for 2008 at LLNL.

The TeraGrid project consists of a total of three individual experiments or climate runs. The allocation will be used by approximately 200 jobs that will consume 5,800 to 6,000 cores for 24 wall-clock hours. Currently, CCSM requires a minimum of 2 GB per MPI task due to legacy defects in the design of its I/O subsystem, meaning that only a single process does I/O, and that if an improved method is used less memory per core would likely be required. Each job will read approximately 80 GB of data from 30 files on disk that range in size from 10 MB to 25 GB. Each job will write ~1 TB of data to 180 files that range in size from 100 MB to 25 GB. Each job will archive ~420 GB of data. Currently, users also export about 420 GB of data per day from NICS to NCAR for analysis. The current approach is only made possible by transfer rates enabled by high bandwidth wide area networks and high performance protocols like gridFTP. There are plans to refine the analysis approach in the future to limit the amount of data transferred back to NCAR.

Production runs often exhibit a large degree of variability in Lustre filesystem performance on the Kraken XT5 at NICS. Specifically, the time to write output files varies by a factor of 18x, from about 5 MB/s to about 92 MB/s. Because of this variability, the overall cost of disk I/O is about 23 percent of the total cost of the simulation versus the expected 10%. This excessive variability complicates the ability to accurately predict the project's computational resource requirements.

Research in this area was limited by a perceived access to resources. An initial estimate suggested 100M hours was needed to complete the research goals. However, it was thought that this would be viewed as being excessive, and the actual request was limited by deferring one of the four planned runs and by reducing the number of cores used by the jobs to reduce the overall cost. Unrestrained by computational resources, this work could easily use ~100M hours per year. The jobs would use 20,000 to 56,000 cores, run for 24 hours and require 2 GB/core. In these runs it appears that the ocean component would dominate the runtime.

Potential improvements include addition of parallel I/O to all components; increase in resolution of existing component models; and the addition of more scalable component models, in particular, a more scalable atmospheric dynamical core. These changes would result in a 5-10x increase in the number of files generated as well as a 5x increase in the size of 20 percent of the files.

One of the important challenges for larger-scale parallelism will be scalability of POP. This code uses a conjugate gradient solver with a global reduction operation (in MPI) for which both hardware and software effects can hinder scalability. This may be significant impediment to this project's productivity on 10K-1000K PE systems.

Another issue of concern is overall system stability. Stability issues affect the ability to do development work and complete very long-running production jobs. For example, the project requires a total of three runs to be executed in succession that could use 6,000 cores for approximately 7 months. Sustained system instability greatly impacts ability to achieve scientific objectives.

6.2.6.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	35,000,000	100,000,000
Parallel Concurrency	5,800	6,000 – 30,000
Wall Hours per Run	24	24
Aggregate Memory	11,600 GB	40,000 GB
Memory per Core	2 GB	2 GB
I/O per Run	1,044 GB; .9 - 30 GB per file	1,000 GB; 30 GB per file
On-Line Storage Needed	20 GB, 200 Files	20 GB, 200 Files
Data Transfer	414 GB	
Archival Storage	200 GB, 24,000 Files	200 GB, 24,000 Files

6.2.6.3.2 Support Services and Software

We require gridFTP support.

6.2.6.4 Emerging HPC Architectures and Programming Models

All components models support MPI, OpenMP and hybrid MPI/OpenMP.

7 Environmental Science

7.1 BER Environmental Science Overview

The Environmental Remediation Sciences Program advances fundamental science to understand, predict and mitigate the impacts of environmental contamination. One of the most challenging problems in environmental remediation involves hazardous materials that have leached into the subsurface and are at risk of being more widely dispersed by the flow of groundwater through contaminated areas. Scientifically rigorous models of subsurface reactive transport that accurately simulate the movement of contaminants across multiple length and temporal scales are required for better understanding the movement of subsurface contamination. NERSC has played and will continue to play a vital role in enabling modeling and simulation for this key BER mission driver.

Important biogeochemical processes (e.g., microbial respiration) are best understood at very small scales, typically ranging from molecular to cellular level, with time scales of minutes to days. However, predicting phenomena in aquifers requires very large simulations typically ranging from meters to kilometers and time scales of months to years or even centuries. This problem is aggravated by the variability of natural subsurface properties that exist across the broad spectrum of spatial and temporal scales.

The Scientific Discovery through Advanced Computing (SciDAC) "Scaling the Challenges in Subsurface Simulations" project will use HPC to customize and apply existing multiscale hybrid modeling methodologies to subsurface science to advance both scientific understanding and create a predictive capability that is applicable to field-scale problems. This supports a key DOE/BER long-term measure for Environmental Remediation, which is to provide sufficient scientific understanding of these multi-scale phenomena such that contaminated DOE sites can incorporate coupled physical, chemical and biological processes into decision making for environmental remediation and long-term stewardship.

7.2 Environmental Sciences Case Studies

7.2.1 Hybrid Numerical Methods for Multiscale Simulation of Multicomponent Subsurface Biogeochemical Reactive Transport

Principal Investigator: Timothy Scheibe, Pacific Northwest National Laboratory (PNNL)
Contributors: Bruce Palmer, Alexandre Tartakovsky, Yilin Fang (PNNL); Paul Meakin, Idaho National Laboratory

7.2.1.1 Summary and Scientific Objectives

The goal of this SciDAC Science Application is to characterize and model natural subsurface heterogeneity and its impact on biogeochemically reactive transport in groundwater systems. We seek to develop an integrated multiscale modeling framework that can directly link different subsurface flow, transport, and reaction process models at continuum, pore, and sub-pore scales. The challenge is in addressing spatial heterogeneity in the subsurface materials, the effect of multi-phase, multi-domain, coupled processes involving water, air/gas, non-aqueous phase liquids (oils, solvents), supercritical fluids (CO_2), and mineral precipitation, and uncertainty quantification in the results. Important applications requiring these complex multiscale, extensive time period simulations (100s-1000s years) include the fate and transport of microbial and other contaminants in aquifers, bioremediation of metal and radionuclides, and geologic carbon sequestration.

7.2.1.2 Methods of Solution

The computational approach involves three focus areas. Pore-scale simulation of fluid flow is done to incorporate and understand fundamental biogeochemical processes. Pore scale basically means a few to a few hundred microns. For this we use the SPH_CCA code, which performs smooth particle hydrodynamic simulations (SPH). The SPH formulation of the hydrodynamic equations uses a random sampling of the continuum hydrodynamic fields by discrete points (the SPH particles). It uses simple explicit time integration methods to generate trajectories of the SPH particles. The code is currently incorporated into the CCA (Common Component Architecture) framework. We also use PARASIM, which incorporates three popular particle simulation methods: molecular dynamics (MD) with embedded-atom and Lennard-Jones potentials, dissipative particle dynamics (DPD), and smoothed particle hydrodynamics (SPH). PARASIM is based on LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) and the Parallel DYNAMO code developed at Sandia National Laboratories. It is written in Fortran 90 and C with domain and force decomposition parallelism via MPI. Another code, TE2THYS, a PNNL (Pacific Northwest National Laboratory) code for arbitrary pore geometry, is also used.

The second focus area is Scale Integration, which arises because of the obvious problem: It is impractical to simulate engineering problems of interest with pore-scale resolution.

Upscaling involves the use of fundamental-scale models to inform larger-scale simulations. STOMP is a general-purpose tool for simulating subsurface flow and reactive transport. Its target capabilities were guided by proposed or applied remediation activities at sites contaminated with volatile organic compounds and/or radioactive material. The simulator's capabilities address a variety of subsurface environments, including nonisothermal conditions, fractured media, multiple-phase systems, nonwetting fluid entrapment, soil freezing conditions, nonaqueous phase liquids, first-order chemical reactions, radioactive decay, solute transport, dense brines, nonequilibrium dissolution, and surfactant-enhanced dissolution and mobilization of organics. STOMP is currently being used to simulate contaminant transport at several sites at the DOE Hanford reservation and other DOE sites.

The STOMP simulator solves the partial-differential equations that describe the conservation of mass or energy quantities in porous media. The simulator has been written with a variable source code that allows the user to choose the solved governing equations (e.g., water mass, air mass, dissolved-oil mass, oil mass, salt mass, thermal energy). Depending on the chosen operational mode, the governing transport equations will be written over one to four phases (e.g., aqueous phase, gas phase, (nonaqueous phase liquid) NAPL phase, ice phase, solid phase).

STOMP applies integrated-volume finite-difference discretization to the physical domain and backward Euler discretization to the time domain. The resulting equations are nonlinear coupled algebraic equations, which are solved using Newton-Raphson iteration. Solute transport, radioactive decay, and first-order chemical reactions are solved using a direct solution technique (e.g., Patankar's power-law formulation, (total variation diminishing) TVD scheme) following the solution of the coupled flow equations.

The third focus area is Field-Scale Simulation, the simulation of complex processes at field sites, such as the Hanford Site 300 area. An important continuum-scale code is PFLOTRAN but this code is not currently heavily used at NERSC.

An alternative method that we are exploring at NERSC is Hybrid Multiscale, in which we couple pore- and continuum-scale models in a single simulation. This is like an AMR approach and is done via the Common Component Architecture (CCA), exchanging info via a CCA coupling component, somewhat like the coupler in a climate simulation.

7.2.1.3 HPC Requirements

Our current studies are primarily on NERSC’s Franklin system and an Opteron/Infiband cluster at PNNL. At NERSC we plan to run several large simulations on the order of 10 million SPH particles to simulate convection and diffusive transport in porous media. Our current estimates are that we can simulate about 1,000 steps in an hour using 500 processors for calculations in this size range. We also would like to investigate larger systems, if possible, and feel that we could do 100-million-particle simulations on Franklin. For calculations in the 500-1,000 processor range and 10 million particles, I/O, currently about 1-10GB per snapshot, is not a huge bottleneck but it will be a problem for

larger problem sizes. We don't track memory usage because we believe it is not significant.

What follows are our estimates for two groundwater simulations, representing what would be large, but hopefully not heroic, studies in the 2015 timeframe.

The first is a continuum Darcy-scale simulation using approximately 1 billion grid cells. The target is an area approximately 1 km square by 100 m deep using a resolution of 1 m in the horizontal direction and 10 cm in the vertical direction. This would be a field-scale simulation of multicomponent reactive transport with sufficient resolution to represent geologic heterogeneity and resolve localized contaminant sources, wells, waste tanks, mixing zones, etc. We could, for example, simulate migration of radioactive contaminants such as Tc-99 leaking from subsurface waste tanks at the Hanford Site 200 Area. A simulation of this type would contain processes such as

- Multiphase flow
- Solute transport (advection/dispersion)
- Multi-rate mass transfer
- Surface sorption on solid phase
- Multicomponent chemical reactions including both equilibrium and kinetic reactions, also including heterogeneous reactions (e.g., precipitation/dissolution) and biogeochemical reactions (e.g., microbial-mediated metal reduction)

A simulation of this size would also require about 60 fields defined on the cells, which would require about 480 GBytes of aggregate memory. The simulation would need to execute on the order of 1,000-10,000 timesteps to be useful. Current simulation codes can do about one timestep per 15 minutes of wallclock time with a million cells per core. This works out to 250 hours for a 1,000-step simulation on 1,000 processors. For 100,000 processors and 10,000 steps, this simulation would require 25 hours of wallclock time, assuming perfect scaling. This estimate will vary greatly depending on the physics included in the simulation and the algorithms used; linear and non-linear algebraic systems solvers must scale close to order N to allow efficient solution of very large problems (see above discussion on MPI_Allreduce scaling).

Second, we consider an SPH simulation of pore-scale flow that includes physical processes such as multicomponent equilibrium and kinetic chemical reactions, heterogeneous reactions (e.g., precipitation/dissolution) and multiphase flow such as air/water or water/non-aqueous phase liquids. A simulation of this size would model a volume of porous media containing on the order of 10,000 grains, which begins to approach the number required for a representative volume simulation in three dimensions. Such a simulation would require about 100 particle fields and would represent an aggregate memory requirement of 1 TByte. A basic simulation of flow in a system of this size would require about 200,000 timesteps. Current simulations of 14 million particles on 2,048 cores are using about 20 hours for 60,000 timesteps, so this would amount to about 100 hours of wallclock time on 100,000 cores, assuming perfect scaling. Additional improvements might be achievable using better implementations of

the communication model and via multithreaded implementations of the force calculation, so that the 100-hour figure might be reduced to 50 or even 20 hours.

7.2.1.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	700K	2 Million
Parallel Concurrency	2,000	10,000
Wall Hours per Run	24	24
Aggregate Memory	1 TB	5 TB
Memory per Core	0.5 GB	0.5 GB
I/O per Run	100 GB	500 GB
On-Line Storage Needed	1 GB / 800 Files	1 GB / 800 Files
Data Transfer		
Archival Storage	3 GB / 10,000 Files	3 GB / 10,000 Files

7.2.1.3.2 Support Services and Software

Our codes require efficient parallel solvers via PETSc and other solver libraries, Global Arrays, and high performance parallel IO libraries such as pNetCDF and HDF5. We believe that resources dedicated to software (libraries) and operating systems must be commensurate with resources dedicated to hardware.

Four crucial elements of this work are: (1) advanced high-performance component architectures; (2) a component-based workflow environment, presently Kepler, to facilitate data provenance capture, I/O file tracking, visualization, and job submission / monitoring; (3) scalable, high performing parallel I/O libraries for writing to large, shared files; and (4) scalable, high-performance visualization tools to display the output of simulations. Regarding component architectures, note that CCA requires support for shared libraries. The CCA support group is exploring a static option but this eliminates runtime configuration, which we use to swap out different SPH components, or different chemistry components for different reactions in our continuum scale code. Regarding I/O and visualization, although on-the-fly analysis will be an option for reducing output data volumes, it will still be desirable to output at least some fraction of results for visualization.

We also find it useful to have good profiling tools such as TotalView, IPM and TAU, that support detailed examination of the behavior of code segments and can organize results in human-understandable form. It will probably become increasingly necessary to do debugging and profiling on very large processor counts. Finally, we note that our data archive process needs to be incorporated into the job execution services (e.g. write data to disk and then move it to archive before job execution completes) to eliminate possible flooding of the disk.

7.2.1.4 Emerging HPC Architectures and Programming Models

We do not have a comprehensive strategy for dealing with multicore architectures at this time. However, we are in close contact with the Global Arrays development team, which is looking at the issue of multicore programming models and will be following developments closely. Also, the component strategy that we have been following for developing our codes will provide a lot of flexibility for incorporating new communication paradigms into our software as they are developed. It is clear, though, that progress in our field will require advances in linearly or near-linearly scaling solvers, adaptive algorithms (hybrid multiscale code coupling), adaptive mesh refinement, and remote parallel visualization (ray tracing).

8 Biological Systems Science

8.1 BER Biological Systems Science Overview

The Biological Systems Science Division manages a diverse portfolio of fundamental research and technology development to achieve a predictive, systems-level understanding of complex biological systems in support of DOE missions in energy, the environment, and carbon sequestration. The Division's Genomics:GTL program seeks to develop the computational capabilities and systems needed to predictively design and model biological systems. This program is developing genome-scale technologies needed to understand the function of microbial and plant systems, from proteomics (the large-scale study of proteins, particularly their structures and functions) to metabolomics (the study of the unique chemical fingerprints left behind by cellular metabolism) to regulatory networks to ecogenomics (the study of genetic material recovered directly from environmental samples). Genomics:GTL research also aims to deliver the transformational breakthroughs in basic science that will enable the development of cost-effective, commercially viable technologies for producing next-generation cellulosic biofuels. NERSC is the flagship provider of HPC resources in support of all these efforts.

To fully understand biological systems is to understand biological phenomena in their full complexity. This requires an understanding of networked and responsive biological functions that are both time and context dependent and engaged in multiscale processes. In the area of computational biophysics, BER projects are actively engaged in developing and applying atomistic-molecular to coarse-grained mathematical models of potential energy surfaces, characterizing these surfaces through sampling techniques and finally generating ensemble or time averaged physical properties of biological phenomena. This work rests firmly on theoretical foundations in quantum and classical physics and statistical mechanics in order to overcome challenges in modeling biological systems. In addition, we are also engaged in large-scale molecular dynamics simulations of complex biological macromolecules whose functions impact our understanding of a wide range of energy and environmental mission-driven sciences.

NERSC also supports fundamental research in the redesign of microbial metabolic processes to harness their potential in the conversion of biomass to biofuels with the ultimate goal of relieving our dependency on petroleum products as well as to impact biogeochemistry phenomena. This work requires the sequencing and annotation of complete microbial genomes, elucidation of metabolic pathways, and simulations of biological processes. Simulations run at NERSC are unraveling functional annotations of unstructured proteins from analysis across genomic and structural relationships. This work requires comparison across large datasets as well as dynamical simulations of protein folding into three-dimensional constructs and makes use of advances in machine learning and physics based simulations.

In summary, the HPC requirements for computational biophysics and bioinformatics are those that will enable biological simulations to be performed with both greater accuracy and complexity so as to guide experimentation that leads to discovery of new and emergent properties arising from a systems view of biology. Advancing our ability to predict an organism's phenotype from a genomic sequence requires an integration of computational modeling, algorithm and software development with new advances in hardware architecture. We must therefore sustain and steward efforts in parallel algorithm and software development, provide computing resources that devise new strategies for data archive/analysis and promote new methods for data visualization.

8.2 Molecular Dynamics

Principal Investigators: Paul Adams, Lawrence Berkeley National Laboratory; Ioan Andricioaei, University of California, Irvine; Teresa Head-Gordon, University of California, Berkeley and Lawrence Berkeley National Laboratory; M. Karplus, Harvard University; Jeremy Smith, Oak Ridge National Laboratory;

Contributors: Dylan Chivian, Lawrence Berkeley National Laboratory; Jingzhi Pu, Harvard University

8.2.1 Overview

"*The structure and function of biomolecular machines are the foundation on which living systems are built. Genetic sequences stored as DNA translate into chains of amino acids that fold spontaneously into proteins that catalyze chains of reactions in the delicate balance of activity in living cells. Interactions with water, ions, and ligands enable and disable functions with the twist of a helix or rotation of a side chain. The fine machinery of life at the molecular scale is observed clearly only when frozen in crystals, leaving the exact mechanisms in doubt. One can, however, employ molecular dynamics simulations to reveal the molecular dance of life in full detail*. " – Klaus Shulten, James C. Phillips, Laxmikant V. Kale, and Abhinav Bhatele.[5]

Molecular simulation in computational biology rests firmly on the theoretical foundations of quantum, classical and statistical mechanics, and faces the primary challenge of overcoming (often great) technical obstacles to make them successful in biological modeling. Despite these challenges, there are a vast number of biological problems for which molecular simulations will have a major impact on our understanding on a wide range of energy and environmental missions for the Office of Science. These problems include the redesign of microbe metabolism for environmental bioremediation of the nation's most contaminated sites, design of new macromolecules and complexes to aid in the conversion of biomass to biofuels to relieve our dependence on oil, and annotation of

[5] Klaus Schulten, James C. Phillips, Laxmikant V. Kalé, and Abhinav Bhatele. Biomolecular modeling in the era of petascale computing. In David Bader, editor, *Petascale Computing: Algorithms and Applications*, pp. 165-181. Chapman and Hall/CRC Press, Taylor and Francis Group, New York, 2008.

the ~1/3 of microbial and plant genomes that involve unstructured proteins to realize the fruit of the genome sequencing efforts at the beginning of this century.

Molecular dynamics (MD) simulations involve calculating averaged properties from finite-length trajectories. These simulations numerically integrate Newton's equations of motion at very short (~1 fs) timesteps in order to evolve a molecular system of interest in time to generate equilibrium averages or kinetic information. The underlying molecular dynamics engine is a particle-based algorithm that solves Taylor expansion approximations to Newton's equation of motion. There are two levels of problem granularity that makes this algorithm well suited for parallelism. The rate-limiting step for these simulations is the evaluation of empirical energy and forces for N particles due to the long-range Coulombic interactions that are modeled through the Ewald summation. The most common forms of those energies and forces map well onto a fine-grained parallelization. Overlayed on top of this fine-grained parallelization is another layer of coarse-grained parallelization involving the replica exchange sampling algorithm, which runs M independent simulations (each at a different temperature). These involve infrequent short communication to swap state information (position and velocities of all atoms). The basic architectural characteristics supporting this work, therefore, are fast processing speed, generous data cache capacity, and low-latency MPI message passing.

Typical simulations periodically replicate the system in three spatial dimensions, and this approach divides the Coulombic interactions into a short-range part that is evaluated in real space (as a direct sum over atomic positions) and a long-range part evaluated in reciprocal space. For system sizes beyond 10^3 atoms, new formulations of Ewald algorithms have largely reached their crossover from N^2 scaling to $N\log N$ scaling making system sizes of tens of thousands of atoms a reasonable proposition on the most advanced supercomputers. This rate-limiting step must be evaluated 10^6 to 10^{15} times for statistical convergence with the ability to study many, many different biological systems.

Continuing increases in high performance computing technology have rapidly expanded the domain of biomolecular simulation from isolated proteins in solvent to complex aggregates, often in a lipid environment. Such systems routinely comprise 100,000 atoms, with several published simulations exceeding 1,000,000 atoms, and a target peta-scale MD Challenge Problem has been defined involving 100 million atoms. Calculations involving 10^5 particles are currently feasible with the use of existing simulation capabilities on 10^3-10^4 processors. However, studying the function of even the simplest biomolecular machines requires simulations of 100 ns or longer. Because the numerical algorithms are well understood, longer trajectory runs are easily being deployed but wallclock time for the simulation remains a significant bottleneck using processors currently employed in most general-purpose center resources.

BER-allocated projects employing molecular dynamics at NERSC have a variety of science targets but the underlying methods and the resulting center requirements are similar. In what follows, we present vignettes for some these projects as a single case study. Two common themes are the need for long-running simulations despite relatively ample parallelism and the interplay between molecular dynamics simulations and a wide

variety of experimental measurements to gain complete understanding of the physical system. This is especially true in cases involving unstructured proteins or ensembles of structures where experimentation may yield ambiguous results because of population diversity.

A separate case study is presented for a project called "Molecular Dynameomics." While this project also uses molecular dynamics as its fundamental method of physical exploration, the project as a whole has additional NERSC requirements by virtue of its need to compare vast sets of MD simulation results with data from other sources.

8.2.2 Molecular Dynamics Case Studies

8.2.2.1 Molecular Simulations for the Joint BioEnergy Institute

The Joint BioEnergy Institute (JBEI) is designed to address roadblocks in biofuels production and to create the transformational discoveries needed to convert the energy stored in lignocellulose into renewable biofuels. Work at NERSC will involve developing a computational model to enable predictive tailoring of pretreatments to specific biomass types using multi-scale, multi-physics computational approaches. Initially, focus will be on biomass under pretreatment conditions and will consider the macroscopic processes of transport, mechanical deformation, and tissue degradation that arise from microscopic processes at cellular and molecular length scales. The work will utilize GROMACS and LAMMPS.

Pioneering classical and *ab initio* molecular dynamics studies now exist on prototypical imidazolium ionic liquids, showing that parameterizations schemes for empirical force fields are in place to explore problems involving molecular mechanisms of pretreatment of biomass degradation. There is growing consensus that fixed-charge force fields lack the accuracy necessary for these very complex and glassy systems, especially when water is introduced as a co-solvent for "one-pot" cracking and hydrolysis solutions. Systems of this sort will require much more computationally expensive polarizable potentials such as the AMOEBA force field implemented in TINKER and AMBER, in which parameterizations of the ionic liquids [Bmim]Cl, [Emim]Cl, [C4mpy]Cl, and/or AmimCl, exist. Furthermore, TINKER has just released the AMOEBA nucleic acid force field that will allow modeling of lignocellulose (sugar) monomers. This comprehensive set of polarizable force fields, as computationally demanding as they may be, are an absolutely necessary level of modeling for these mixtures involving ionic liquids.

8.2.2.2 Molecular Dynamics Simulations in Bioenergy and Bioremediation

Molecular simulation is required to obtain an understanding of the structure, dynamics and degradation pathways of extended cellulosic and lignocellulosic biomass materials. The physical properties of lignocellulosic biomass thus derived can serve as a basis for interpreting an array of biophysical experiments. This combination of simulation and

experiment will lead to an understanding of biomass recalcitrance to hydrolysis, and thus will aid in developing a strategy as to how rationally to overcome the resistance.

A common theme in all these MD simulations is the long execution times required for useful simulation of realistic molecular systems. With one order of magnitude more computing power beyond what is available today, current 100ns-timescale simulations could be extended to 1 microsecond. This would enable the understanding of biomass recalcitrance to hydrolysis, e.g., illuminating the properties of lignin aggregation on cellulose surfaces. It would also allow the use of more sophisticated models of the ionic liquids and biomass constituents at the 100-ns timescale.

With 100 times more computing power, 10 microseconds can be simulated. Ligand binding to proteins would become accessible to molecular dynamics simulation without further approximation, enabling considerable advances in the development of enzymes for biological problems such as biofuel production, bioremediation, and medicine. We would begin reaching the timescales necessary to study the glassy nature of ionic liquids using polarizable models to understand their biomass degradation mechanisms.

At 1000 times, *ab initio* protein folding should become possible, revolutionizing many areas of medicine and biology, including bioenergy and bioremediation.

8.2.2.3 Molecular Dynamics Simulations in Protein Dynamics

Molecular dynamics simulation of mercury-DNA complexes used in conjunction with X-Ray scattering experiments, will help reveal the long-distance communication pathway between mercury binding sites and protein-DNA interfaces, which is vital for understanding biogeochemical and molecular mechanisms controlling transformation of this key environmental contaminant. This work uses the publically available GROMACS code as well as NAMD, which has been shown to have very good weak scaling. It uses the Fast Fourier Transform-based Particle Mesh Ewald method for non-bonded electrostatic interactions.

As part of a desire to understand biological phenomena in their full complexity, studies are also examining the energetics and time evolution of the interaction of DNA and RNA with proteins and other protein-like or protein-based structures. Such simulations may involve both an equilibrium and non-equilibrium state; hence, simulations may have to run for long times (100-200ns of simulation time) just to get to the time period of interest. Sometimes simulations running to 450 ns are required for a complete free energy curve of a single complex; plus, multiple complex structures are often proposed. Although these simulations use NAMD, which scales well, multi-million resource hours are required in a given year for a full agenda of work.

Knowledge of ensembles of peptides on the aggregation pathway from monomer to larger structures is critical for understanding aggregation outcomes in vitro. Nearly one-third of the sequences in the human genome involve unstructured proteins, ones for which no distinct single tertiary structure exists. Another project seeks to develop an

understanding of the entire aggregation process that ultimately leads to specific known structures, from monomer through intermediate oligomer structures. This involves understanding whether structure in the monomeric peptide is well defined enough to promote ordered stable oligomers. By necessity this work yields results that allow detailed interpretation of Nuclear Magnetic Resonance (NMR) observables from structural ensembles of disordered systems – a key result.

This work uses the Sander component of AMBER version 10.0, a code largely written in Fortran77/90 with MPI. It uses accelerated convergence algorithms, along with the most recent generations of polarizable protein and water force fields.

The following example is meant to be illustrative of the computational requirements for these kinds of simulations. A current investigation includes two monomer sequences (two poly-peptides in water, about 25,000 atoms), and requires 50 ns of simulation at each of 40 temperatures. Two independent trajectories are needed to ensure proper convergence. Using high-quality non-polarizable force fields, this requires the following total number of hours:

13,040 hr (50ns MD) x 40 (T replicas) x 2 (independent simulations) =1,043,200 hrs,

and for the two monomer systems this is ~2.1M hours.

Anticipated future studies might involve simulated structural characterization of additional proposed oligomeric sets of species that we believe will increase the computational size of the problem by over an order of magnitude. When polarizability is used to improve the theoretical model (necessary for ionic liquids) an order of magnitude increase in computational resources relative to the fixed charge simulations will be required.

Additionally, workflow and turnaround time for such simulations are strongly influenced by the way queue policies are set and the resulting turnaround time for jobs to start. Our optimal workflow requirements would be met by having the ability to run uninterrupted trajectories over days as opposed to hours, with access to any number of processors, and including a mechanism for auto-job restart to accommodate typical hardware failures. At present, we are limited by short queues.

8.2.2.4 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	14.6 Million	~150 Million
Parallel Concurrency	1,664	15,000
Wall Hours per Run	235	Queue limit
Aggregate Memory	6.6 TB	~60 TB
Memory per Core	Typically ~2-4GB core	Typically ~2-4GB core
I/O per Run	Output involving: #atoms x 6 double precision numbers	Scale with # atoms and simulation trajectory length

	(Cartesian positions and velocities) x every 10-100fs	
On-Line Storage Needed	Typically all of the above files for post analysis	
Data Transfer	All files produced as above	
Archival Storage	Large trajectory files	

8.2.2.5 Support Services and Software

Unlike other areas such as climate or fusion, that have a much more single-purpose community science goal, the biomolecular simulation community is diverse in terms of the scientific grand challenges that can be accomplished by molecular dynamics, Monte Carlo, and other such techniques. Because MD models, codes and algorithms have reached a high level of maturity that could allow them to solve many important biological problems across a large user base, it would be exciting and productive to create the supercomputer analogy of a "beamline" facility in which dedicated simulation hardware is devoted to biomolecular simulation users who have allocations continuously for one-two weeks per year. This would require developed end user stages with support of the community simulation codes themselves, plus all supporting software, including LAPACK, ScaLAPACK, FFTW, data analysis support of Gnuplot, Grace, ImageMagick, Mathematica, and visualization programs like Rasmol and VMD.

8.2.2.6 Emerging HPC Architectures and Programming Models

There are many codes used by the computational biophysics community that execute essentially the same rate-limiting MD kernel, but have different levels of functionality, degrees of parallelism, and/or code availability. These include AMBER, TINKER, NAMD, CPMD, GROMACS, LAMMPS, CHARMM, DLPOLY, and in-house codes from some research groups, such as Daggett Group's *il*mm. Both MPI and OpenMP are already effectively used to best exploit distributed and shared memory architectures.

The use of special accelerators or co-processors to offload some of the time-consuming code for MD codes has shown great promise. Graphics Processing Unit (GPU) acceleration offers substantial improvement in simulation time over a single general-purpose CPU. GPU hardware is less expensive than general-purpose CPUs for the same level of performance and generally offers the potential for substantial parallelism via functional unit replication. New software environments offer some promise for programming graphics processors in higher-level languages, although portability remains a significant challenge. The availability of GPUs has also stimulated the development of new/enhanced MD algorithms, such as those used to calculate electrostatic potential maps for cutoff pair potentials. However, there are still limitations to the features that can be ported to GPUs that affect the quality (in terms of precision) of some simulations. Furthermore, data transfer between the CPU and the coprocessor can limit performance.

Parts of NAMD, in particular VMD, its simulation setup and trajectory analysis code, have already been enabled to use GPUs for accelerating computation. NAMD is file-compatible with AMBER, CHARMM, and X-PLOR and is distributed free of charge with source code. Several published NAMD simulations have exceeded 1,000,000 atoms. One of the most time consuming calculations in a typical molecular dynamics simulation is the evaluation of forces between atoms that do not share bonds. The ~twenty-fold acceleration provided by GPUs decreases the runtime for the non-bonded force evaluations such that it can be overlapped with bonded forces and PME long-range force calculations on the CPU. These and other CPU-bound operations must be ported to the GPU before further acceleration of the entire NAMD application can be realized.

ACEMD is a proprietary MD software package that can read CHARMM/NAMD and AMBER files and is designed to run on Nvidia® GPUs. There is also ongoing work to port LAMMPS, GROMACS, and AMBER to GPU-based systems. One project, Open Molecular Modeling, seeks to develop an Application Programming Interface for MD simulations so that MD software developers can use standard MD libraries without knowledge of the underlying hardware it runs on; currently both ATI and NVIDIA GPUs are supported.

8.2.3 Molecular Dynameomics

Principal Investigator: Valerie Daggett, University of Washington
Contributors: David Beck, University of Washington

8.2.3.1 Summary and Scientific Objectives

The goal of the Molecular Dynameomics project is to perform native (i.e. biologically active) state and folding / unfolding molecular dynamics simulations of 1641 proteins. These proteins were selected as representatives of every known protein fold / topology type. From the output of these simulations we will construct a database comprised of molecular dynamics (MD) structures for representatives of all protein folds including their unfolding pathways. This database will complement the experimentally determined structural data cataloged in the Protein Data Bank (PDB). The PDB has been a tremendously useful repository of experimentally derived, static protein structures that has stimulated many important scientific discoveries. However, in vivo, proteins are mobile and there is a larger universe of knowledge to be tapped regarding their dynamics. Thus, there is a need for a 'Dynameomics' database of simulations of the approximately 1,641 known non-redundant folds. The protocols employed in these simulations have been developed over the last 15 years in our lab. With continued access to DOE resources, we will be able to simulate all of our targets.

Using data resulting from the MD simulations, we will identify patterns and general features of transition, intermediate and denatured states to improve structure prediction algorithms. Structure prediction remains one of the elusive goals of protein science. It is necessary to successfully predict native states of proteins, in order to translate the current deluge of genomic information into a form appropriate for better functional identification of proteins and for drug design. This is a data-mining endeavor to identify similarities and differences between native and unfolded states across all secondary and tertiary structure types and sequences. This represents our immediate scientific goal for the data resulting from the Dynameomics project; however, as with the PDB after its conception, there will certainly be much more to come of it and areas of inquiry by outside users that we cannot anticipate.

8.2.3.2 Methods of Solution

Our group has developed *in lucem Molecular Mechanics* (*il*mm), a scalable parallel molecular mechanics code. *il*mm uses a classical potential model that explicitly represents solvent and solute atoms. In the simulation the cutoff in the potential calculation must be typically at least 8 Å in order to accurately reproduce experimental observations. In addition, a biophysically correct simulation requires a time-step for numerical integration small enough to accurately represent the trajectories. With the methods employed in *il*mm, we commonly use a 2-femtosecond timestep. As such, the final contribution to computational expense is the highly repetitive nature of the calculations: 1 picosecond of MD requires 500 iterations of the energy function calculation and integration cycle. A microsecond of simulation time requires 500,000,000 iterations.

Sampling in molecular simulations is almost always the most difficult aspect to address – that is, how much simulation time and / or how many simulations are required to accurately sample the event(s) of interest. Many simulations are attempting to document processes that occur on timescales of 100s or 1000s of nanoseconds. As such, we are constantly pushing the boundaries of the MD timescale to extend our simulations. Mathematically, the standard deviation of means varies as $1/N^2$ where N is the number of samples / timesteps, such that longer simulations can provide more precise results.

With the proliferation of inexpensive dual and quad CPU computing platforms, parallel computing has become an attractive solution to reducing the real-world time required for computationally intensive tasks. *il*mm has an optimized force field evaluation that is designed specifically for clusters of multi-processor (i.e. SMP) nodes.

8.2.3.3 HPC Requirements

We have completed simulations of about 800 folds for 100 microseconds. As we simulate proteins from the list of populated folds, the representative proteins become larger, more complex and require more computational time to simulate with molecular dynamics. Small protein systems (of about 50 amino-acids), in all-atom explicit solvent systems (~10,000 atoms) can require as few as 500 hours (NERSC MPP Hours) per 30-ns simulation. Each protein requires six simulations, one as a control at biological temperatures and five more for unfolding. Medium sized systems (150 amino-acids with ~ 50,000 atoms) may require 1500 hours per simulation. The largest systems of 2,100 amino acids, 500 lipid molecules and over 750,000 atoms need 31,000 hours per simulation. Most of the proteins remaining to be simulated fall into the final two size classes (medium and large). We have also determined by extensive analysis of simulations performed at NERSC that some proteins would benefit from better sampling via longer simulations. New targets are added frequently in response to new experimental structures and new interests in biofuels, bioremediation, and disease interests.

Currently, disk space and memory bandwidth are limiting factors for our project.

Moving forward, new HPC challenges arise in the area of analytical computing when we consider the 100+ TB data set that will be produced. The HPC simulation community is facing the problem of how to manage and analyze massive ensembles of data. The computational requirements of analysis are rapidly becoming as significant as simulation. We believe that relational database solutions are an ideal way to leverage existing technologies for delivering precise streams of data to a large number of analytical consumers. We would like to see NERSC embrace analysis as a part of supercomputing.

8.2.3.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	8 Million	4-6 Million
Parallel Concurrency	1,000	Many 1,000s
Wall Hours per Run	288	300
Aggregate Memory	768 GB	4 TB
Memory per Core	<1 GB	4 GB
I/O per Run	1.5 TB	12 TB
On-Line Storage Needed	6 GB / 8500 Files	100 GB / 100,000 Files
Data Transfer	5 TB / 30 days	

8.2.3.3.2 Support Services and Software

Our project requires that computational systems and support servers have access to the GNU Scientific Library, the PERL scripting language, the rsync utility, and the gnuplot analysis tool.

The next challenge will be analyzing the ensemble of 100s of TB of data as a whole. This is now not only a problem of HPC simulation but of HPC analytics.

We need virtualization to support our GrayWulf analytics techniques, database instances for GrayWulf & tools like Dryad, and possibly map/reduce implementations.

8.2.3.4 Emerging HPC Architectures and Programming Models

Our simulation codes are well positioned for the multi- to many-core transition and the return of SIMD. *il*mm has an optimized force field evaluation that is designed specifically for clusters of multi-processor nodes or many-core processors using threads and OpenMP.

8.3 Bioinformatics and Bioengineering Case Studies

Bioinformatics is the application of information technology to the field of molecular biology with the goal of increasing our understanding of biological processes. Its focus is on developing and applying computationally intensive techniques (e.g., data mining, machine learning algorithms, and visualization) to achieve this goal. Major research efforts in the field include sequence alignment, gene finding, genome assembly, protein structure alignment, protein structure prediction, prediction of gene expression and protein-protein interactions, genome-wide association studies and the modeling of evolution.

Bioengineering is the application of engineering principles to address challenges in the fields of biology and medicine. Biological engineering applies principles to the full spectrum of living systems, including molecular biology, biochemistry, microbiology, pharmacology, protein chemistry, cytology, immunology, neurobiology and neuroscience. Bioengineering deals with disciplines of product design, sustainability and analysis to improve and focus utilization of biological systems.

In this report we present two case studies describing projects that have the potential to make significant advances in our understanding of biological systems.

8.3.1 Microbial Genome and Metagenome Data Processing and Analysis with the IMG Family of Systems

Principal Investigator: Victor M. Markowitz, Lawrence Berkeley National Laboratory
Contributors: Natalia N. Ivanova and Nikos C. Kyrpides, Genome Biology Program, DOE Joint Genome Institute

8.3.1.1 Summary and Scientific Objectives

The Integrated Microbial Genomes (IMG) system aims to contribute to the goal of improving overall quality of microbial genome data for the scientific community. IMG is a data management facility and associated software suite that serves as a tertiary community resource for comparative analysis and annotation of all publicly available genomes from three domains of life in a uniquely integrated context. A rapidly increasing number of microbial genomes and metagenome samples are sequenced by organizations worldwide, undergoing similar annotation procedures, and eventual inclusion into public genome data resources[6]. While the combined validation and curation procedures of various data resources improve the quality and completeness of microbial genome annotation[7], erroneous or incomplete annotations are often carried over into public resources and are difficult to correct. This problem is compounded by the rapid increase in the number of sequenced microbial genomes and metagenomes with incomplete and rarely curated annotations. Tertiary microbial genome data resources such as the IMG family of systems aim to provide high levels of data diversity in terms of the number of genomes and metagenomes integrated in the system from public sources, data coherence in terms of the quality of the gene annotations, and data completeness in terms of breadth of the functional annotations. Such a data context is critical for multi genome and metagenome comparative analysis used in the functional characterization of microbial genomes and metagenomes, in particular for reconstructing the metabolic network of an organism or community of microbes using its sequence and the information about metabolic pathways and networks existing in this and other organisms or communities[8]. IMG's main goal is to support the analysis of the genomes sequenced at DOE Joint Genome Institute.

8.3.1.2 Methods of Solution

The **IMG family of systems** consists of:

1. **IMG** (http://img.jgi.doe.gov) which provides support for the comparative analysis of isolate genomes. As of April 2009, there were 4,890 genomes in IMG: 1,284

[6] Markowitz, VM. (2007) Microbial genome data resources, *Current Opinion in Biotechnology* **18**(3), 267-272.

[8] Ivanova NN & Lykidis A. (2009) Metabolic reconstruction. *Encyclopedia of Microbiology,* Elsevier: 607-621.

bacterial, 59 archaeal, 49 eukaryotic, 2,524 viruses and 924 plasmids, with a total of about 5.5 million genes.

2. **IMG/M** (http://img.jgi.doe.gov/m) which provides support for the comparative analysis of metagenomes in the reference context of all isolate genomes in IMG together will all available GEBA genomes[9]. As of May 2009, there were 65 metagenome datasets in IMG/M as part of 21 studies, with a total of about 2.6 million genes.
3. **IMG/ER** (http://img.jgi.doe.gov/er) and **IMG/M ER** (http://img.jgi.doe.gov/mer), which provide support for the functional annotation review and curation by individual scientists or groups of scientists of so called "private" microbial genomes and metagenomes, respectively, prior to public release. Genomes undergoing curation in **IMG/ER** are integrated with all publicly available genomes in IMG. As of April 2009, there were 111 GEBA genomes and 118 private genomes in IMG ER, with a total of about 1 million genes. Metagenomes undergoing curation in **IMG/M ER** are integrated with all publicly available genomes in IMG and all publicly available metagenomes in IMG/M. As of May 2009, there were 213 private metagenome datasets in IMG/M ER as part of 40 studies, with a total of 5.1 million genes.

Users access IMG via a web browser that connects to a remote Apache web server running the IMG Web Data Explorer application. The application is implemented using Perl 5.8.x and employs the GD package for graphics. Exploration Viewers and tools handle the data exploration operations, such as gene search and genome browser, and provide support for running external tools such as BLAST, ClustalW, and JalView. The IMG back end (data server) consists of the IMG warehouse implemented with the Oracle 9i database management system, BLAST databases for similarity searches against NR, SwissProt, Pfam, and IMG genes, and auxiliary data files that contain scaffold DNA sequences or KEGG map images. The IMG back end also includes pre-computed statistics and phylogenetic profiles, BLAST homolog results, and other cache data for improving performance, such as pre-computed gene/scaffold/cog mapping data for the ortholog neighborhood viewer and the genome line positions in the phylogenetic genome browser.

8.3.1.3 HPC Requirements

The main **content update cycle** for the IMG systems starts every four months with the update of IMG:

1. A **typical IMG content update** cycle involves the integration of about 350 new genomes with a total of about .6 million genes from one or several public resources, and consists of several stages across 3 months. The **computation** stage consists of the functional annotation of individual genes, identification of pair wise gene

[9] A Genomic Encyclopedia of Bacteria and Archaea (GEBA). See: http://www.jgi.doe.gov/programs/GEBA/pilot.html

relationships (e.g., paralogs, homologs, othologs), identification of chromosomal clusters (gene cassettes) and their conservation across all genomes (conserved gene cassettes). The computation stage of an IMG update cycle involves running various flavors of Blast and HMM protein searches, and takes about 3 weeks on a Linux cluster consisting of 230 cores, 4 GB/core, 500 GB data read/written, with 20 TB on-line storage. The computation stage is scheduled with a strict start and end, whereby delays would affect the subsequent stages of the cycle.

2. The content update for IMG is followed by the update of the public reference genome (**baseline**) part of **IMG ER**. The IMG ER baseline content update starts at the end of the second month of IMG's update cycle, takes about 1 month, and includes 2 weeks of computations on a Linux cluster consisting of 230 cores.
3. The baseline content update of IMG ER is followed by the update of the public reference genome part of **IMG/M ER**, which includes all the genomes in IMG plus all GEBA genomes in IMG ER. This IMG/M ER content update starts at the end of the third month of the IMG's update cycle, takes about 1 month, and includes 3 weeks of computations on a Linux cluster consisting of 230 cores.

About twice a year, IMG/M is updated with newly published metagenomes from IMG/M ER. IMG/M can be viewed as an IMG/M ER data mart, and involves negligible computations. Finally, IMG ER and IMG/M ER have monthly content updates involving new private genomes and metagenomes, whereby each update cycle takes about a week on a Linux cluster consisting of 230 cores.

The main challenge for the next 2-3 years is maintaining a regular 4 month cycle for updating the IMG systems while facing a rapid growth in the number of isolate genomes and a substantial increase in the size and complexity of metagenome datasets in terms of the number of their genes or gene fragments, which are expected to grow from an average of tens of thousands of genes to tens of millions of genes per dataset. Various flavors of Blast and HMM searches will continue to dominate the computations involved in updating the IMG systems.

We expect to address these challenges by increasing the computing infrastructure and by developing new data reduction methods that may help alleviate the growth in the number and size of the datasets.

8.3.1.3.1 Computational and Storage Requirements Summary

	Current	**Next 3-5 Years**
Computational Hours	300,000	
Parallel Concurrency	232	900
Wall Hours per Run	340	340
Aggregate Memory	64	256
Memory per Core	4 GB minimum	16 GB
I/O per Run	500 GB	1500 GB
On-Line Storage Needed	20 GB, 65 million files	40 GB, 1 billion files
Data Transfer		

8.3.1.3.2 Support Services and Software

We would like an experimental testbed for alternative large distributed file systems, like for hadoop.

8.3.1.4 Emerging HPC Architectures and Programming Models

Many bioinformatics problems use polynomial time algorithms (e.g., all-to-all comparisons) although the exact computational complexity varies according to sequence type/length and the nature of the search. However, nearly all involve large data volumes. Computational efficiency of these algorithms is more related to data movement than it is to floating-point arithmetic processing speed. Recent work has demonstrated that there appears to be ample parallelism in many bioinformatics workloads but adaption of some algorithms to heterogeneous processors combining general-purpose cores with Single-Instruction Multiple-Data (SIMD) implementations have revealed limitations in these architectures. Clearly, significant benefit will derive primarily from improvement in data management to support processing large databases.

We would like to test the existing multi-threaded BLAST on a single host with a large (~256GB) amount of memory so that a RAM disk for BLAST databases can be used, with a large number of multiple cores (say 1,000) for fast "on the fly" runtime BLAST for our UI applications. Speeding up the runtime of BLAST and avoiding large amounts of pre-computations and subsequent disk data management problems is one strategy we would like to test out.

8.3.2 Protein and Metabolic Engineering, Combinatorial Network Optimization, and Bioinformatics

Principal Investigator: Costas Maranas, Pennsylvania State University

8.3.2.1 Summary and Scientific Objectives

Our work focuses on development of computational approaches for the elucidation of biological systems and networks and the identification of engineering interventions for enhancing desired properties. Research in protein engineering is focusing on finding ways to computationally prescreen protein hybrids generated through mutation or recombination for their potential to form stable folds with retained/improved functionalities. Research on biopathways focuses on the development of computational tools for effectively querying the performance limits of genome-scale models of metabolism and identifying recombination strategies (i.e., gene knock-ins and knock-outs) that lead to the overproduction of desired chemicals. As size and complexity of biological networks increases this will severely tax the computational performance of the analysis, curation and redesign tools.

This work has four high-level goals:

1. To develop computational methods that will simultaneously bring to bear multiple types of analyses and data (i.e., network connectivity, gene essentiality experiments, metabolomic and transcriptomic data) to automatically assess the quality of genome-scale reconstructions and generate hypotheses for their correction.

2. Using as input genome-scale metabolic reconstructions from Goal 1, we want to develop computational techniques that will enable the largely automated tracking of isotope-labeled atoms. This will provide comprehensive isotope tracking maps to support metabolic flux elucidation through metabolic flux analysis (MFA).

3. We then plan to make use of the developed isotope maps of Goal 2 to estimate metabolic flux values consistent with experimental data and pinpoint what additional measurements are needed to fully resolve all fluxes in genome-scale metabolic models. To this end, we will develop customized global optimization procedures for addressing the presence of nonlinearities in the isotope balance equations and extend the algorithmic base to handle isotopically nonstationary data using dynamic optimization concepts as well as quantify the impact of measurement error.

4. We then want to develop computational tools that will make direct use of flux information from Goal 3 as well as regulatory, thermodynamic or even kinetic information whenever available. The key concept here is that instead of looking for specific engineering strategies one at a time, we seek to classify all fluxes in the metabolic model depending upon whether or not they must increase, decrease, or become equal to zero to meet a pre-specified overproduction target. Additionally, once we identify these pathways, we can explore whether overexpression alone can achieve the

necessary activity, or whether we need to explore computationally engineering substrate specificities and catalytic activity through protein design.

8.3.2.2 Methods of Solution

Third-party application software is typically used, including CPLEX, CONOPT, CHARMM, and Gaussian03.

The CPLEX and CONOPT codes use mathematical optimization (i.e., MILP and NLP) and combinatorial graph analyses. The size of optimization problems is characterized by thousands of binary variables. Graph sizes are also large accounting for thousands of nodes. Parallelism is currently handled by manually segregating computing tasks to different computing nodes.

In-house protein design algorithms utilize optimization formulations (i.e. MILP) as well as molecular mechanics energetic calculations (i.e. binding energy, molecular dynamics) on the order of tens of thousands of atoms. We utilize quantum mechanics calculations (i.e. DFT, MP2) to estimate the ground and transition state energetics of proteins to their substrates. For these calculations, we use MPI for communication between processors.

In-house developed software includes IPRO, OptGraft, GapFill, GrowMatch, and Optknock, which is a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization.

8.3.2.3 HPC Requirements

Our current runs typically use between four and 200 Intel x86 cores in Linux systems with about 2 GB of memory per core and run for 200 to 400 wallclock hours.

Substantially more computing capability at a facility such as NERSC would significantly advance the science we are attempting. For example, in the context of strain optimization for biofuel production we will be able to explore a much larger number of simultaneous genetic manipulations. I n the context of protein design, we will be able to explore a much larger ensemble of amino acid mutations and higher levels of theory in the *ab initio* calculations. Key limitations include computational time due to the NP-hard nature of the underlying mathematical problems. In some cases, memory usage due to the combinatorial explosion of branch-and-bound trees becomes limiting.

The computational biological network community needs to plan in anticipation of the multi-compartment (tissue-specific) models that are currently under development for plant systems as well as community models when multiple microbial species co-exist. This new level of complexity will increase the size of problems by at least an order of magnitude in the near future.

One objective is that we will need to develop techniques to solve flux ranges in an automated and fully parallelized manner. Additionally, the non-linear nature of flux elucidation for large-scale models can benefit from the use of decomposition and a computationally efficient representation of the isotope mappings.

Our in-house protein redesign algorithm Iterative Protein Redesign and Optimization (IPRO) currently has time consuming steps that can be greatly reduced through parallelization. Access to more nodes simultaneously than we currently use can reliably obtain efficient protein redesigns in significantly less time. IPRO currently requires the manual determination of the initial starting candidate. We anticipate that being able to massively parallelize IPRO will allow us to computationally determine redesigns of a large number of candidate structures.

Determining which measurements to make in order to elucidate fluxes using an incidence matrix will require large amounts of memory, even when using sparse matrices. We anticipate scaling will require increased memory usage and efficient branching algorithms.

We are also currently limited solving global optimum problems by the size of our models using the branch and bound technique. Having access to longer-term jobs and more memory would allow the determination of more physiologically relevant models.

9 Participants and Contributors

R. Todd Anderson	DOE / BER	Environmental Science
Anjuli Bamzai,	DOE / BER	Climate Science
Ioan Andricioaei	UC Irvine	Biological Science
David Beck	University of Washington	Biological Science
Lawrence Buja	NCAR	Climate Science
Dylan Chivian	LBNL	Biological Science
John Dennis	NCAR	Climate Science
John Drake	ORNL	Climate Science
Susan Gregurick	DOE / BER	Biological Science
Teresa Head-Gordon	LBNL	Biological Science
Christopher Kerr	GFDL	Climate Science
Lucy Nowell	DOE / ASCR	Program Manager
Costas Maranas	Pennsylvania State University	Biological Science
Victor Markowitz	LBNL	Biological Science
Lee Ann McCue	PNNL	Biological Science
Bruce Palmer	PNNL	Environmental Science
Edward Patton	NCAR	Climate Science
David Randall	Colorado State University	Climate Science
Timothy Scheibe	PNNL	Environmental Science
Yukiko Sekine	DOE / ASCR	NERSC Program Manager
John Shalf	NERSC	NERSC Group Lead
Jeremy Smith	ORNL	Biological Science
David Thomassen	DOE / BER	Chief Scientist
Michael Wehner	LBNL	Climate Science
Christopher L. Wolfe	Scripps Institute	Climate Science
Kathy Yelick	NERSC	NERSC Director
	Editors	
Richard Gerber	NERSC	NERSC User Services
Harvey Wasserman	NERSC	NERSC User Services

10 Acknowledgements

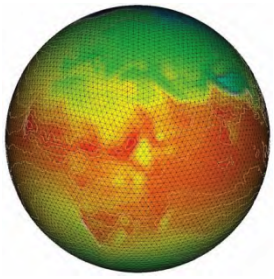
This work would not have been possible without the contributions and participation of those researchers who provided information and attended the workshop. NERSC would also like to thank the BER and ASCR program offices and managers for their help in organizing the workshop and providing insight into the science supported by the BER program. In addition, NERSC Director Kathy Yelick, NERSC Department Head Francesca Verdier, and NERSC Science Driven System Architecture Lead John Shalf provided very helpful advice before, during and after the workshop. A special thank-you goes to ASCR Program Manager Yukiko Sekine who provided the initial impetus for the workshop. We thank Margie Wylie-Petruzziello for her assistance in preparing the document cover.

NERSC is funded by the US Dept. of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) program. Yukiko Sekine is the NERSC Program Manager. ESnet is operated by Lawrence Berkeley National Laboratory, which is operated by the University of California for the US Department of Energy under contract DE-AC02-05CH11231.

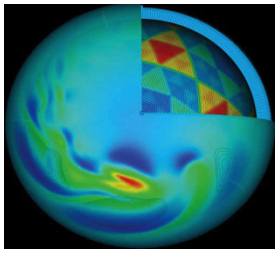
This work was supported by the Directors of the Office of Science, Office of Advanced Scientific Computing Research, Facilities Division, and the Office of Biological & Environmental Research.

This is LBNL report LBNL-2710E, published October, 2009

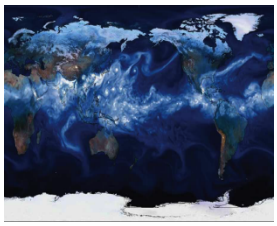
About the Cover



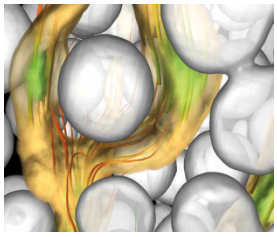
Simulation of a global cloud resolving model (GCRM).
This image shows the surface temperature and the geodesic grid. Image courtesy of Professor David Randall, Colorado State University.



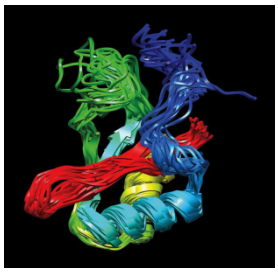
Simulation of a global cloud resolving model (GCRM).
This image is a composite plot showing several variables: wind velocity (surface pseudocolor plot), pressure (b/w contour lines), and a cut-away view of the geodesic grid. Image courtesy of Professor David Randall, Colorado State University.



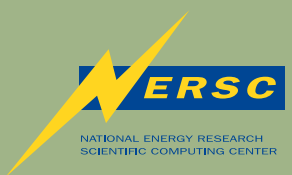
Visualization of CCSM Simulation Data with VisIt showing hurricane formation and evolution. Dataset provided by Michael Wehner (LBNL); visualization provided by Prabhat (NERSC).



Zoomed-in view of a 3D visualization of pore-scale fluid flow computed using the parallel Smoothed Particle Hydrodynamics code developed in the Computational Hybrid Integration of Physical Processes across Scales (CHIPPS) project, Tim Scheibe (Science Application Lead).



This image depicts a native state molecular dynamics simulation of the enzyme RuBisCO, showing a schematic representation of the secondary structure. This enzyme is most abundant protein in leaves and possibly the most abundant protein on Earth. Image courtesy of Valerie Daggett and Marc van der Kamp, University of Washington.


ERSC
NATIONAL ENERGY RESEARCH
SCIENTIFIC COMPUTING CENTER


U.S. DEPARTMENT OF
ENERGY
Office of Science


BERKELEY LAB
Lawrence Berkeley
National Laboratory