DISCLAIMER

This report was prepared as an account of a program review sponsored by the U.S. Department of Energy. Neither the United States Government nor any agency thereof, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Copyrights to portions of this report (including graphics) are reserved by original copyright holders or their assignees, and are used by the Government's license and by permission. Requests to use any images must be made to the provider identified in the image credits.

Ernest Orlando Lawrence Berkeley National Laboratory is an equal opportunity employer.

Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720 U.S.A.

NERSC is funded by the United States Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) program. David Goodwin is the NERSC Program Manager and Lali Chatterjee serves as the High Energy Physics (HEP) 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 High Energy Physics, and the Office of Advanced Scientific Computing Research, Facilities Division.

This is LBNL report LBNL-XXX published YYYY.

Large Scale Production Computing and Storage Requirements for High Energy Physics: Target 2017

Report of the Program Requirements Review
Conducted November 27 - 28, 2012
Rockville, MD

DOE Office of Science

Office of High Energy Physics (HEP)
Office of Advanced Scientific Computing Research (ASCR)

National Energy Research Scientific Computing Center (NERSC)

Editors
Richard A. Gerber, NERSC
Harvey J. Wasserman, NERSC

Table of Contents

1 Executive Summary

The National Energy Research Scientific Computing Center (NERSC) is the primary computing center for the DOE Office of Science, serving approximately 4,500 users working on some 550 projects that involve nearly 700 codes for a wide variety of scientific disciplines. In addition to large-scale computing and storage resources, NERSC provides critical support and expertise to help scientists make efficient use of these resources.

In November 2012, NERSC, DOE's Office of Advanced Scientific Computing Research (ASCR), and DOE's Office of High Energy Physics (HEP) held a program requirements review to characterize HPC requirements for HEP research over the next five years. The effort is part of NERSC's continuing involvement in anticipating future user needs and deploying necessary resources to meet these demands.

The workshop revealed several key requirements, in addition to achieving its goal of characterizing HEP computing. The key findings are:

1. Researchers need access to significantly more computing and storage resources to support HEP mission goals through 2017.
2. Scientists need vastly improved data I/O rates and better facilities for performing data-intensive science.
3. Research teams need to run both large-scale simulations and massive numbers of individual jobs.
4. NERSC must help enable and ease the transition to next-generation architectures.

In addition to these requirements, the review found that there are communities within DOE HEP that are not traditional users of large HPC centers, yet have a profound need for additional computing, storage, and analysis facilities. These "non-traditional" users include researchers on the ATLAS and CMS experiments at the Large Hadron Collider and related theory and those involved with current and planned cosmological sky surveys.

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

The report 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 High Energy Physics Program Mission and Computational Priorities

Lali Chatterjee, DOE Office of High Energy Physics

The High Energy Physics (HEP) program mission is to understand how the universe works at its most fundamental level. This is done by discovering the elementary constituents of matter and energy, probing the interactions between them, and exploring the basic nature of space and time. This quest takes us from micro scales deep inside hadrons, to cosmic scales of millions of light years via a worldwide program of particle physics research.

HEP research probes the universe to understand fundamental particle properties, discover new phenomena and learn about the 'dark universe' through three complementary frontiers - Energy, Intensity and Cosmic Frontiers.

At the Energy Frontier, collider and fixed target experiments create new particles, reveal their interactions, and investigate fundamental forces; at the Intensity Frontier, experiments explore fundamental forces and particle interactions by studying events that rarely occur in nature; and at the Cosmic Frontier, observations and measurements offer new insight and information about the nature of dark matter and dark energy. The strong connections between these key questions necessitate coordinated initiatives across the three complementary frontiers to answer some of the most basic questions about the world around us.

HEP research proceeds along experimental paths - accelerator based and observational, along with theory and computation. Experiments and Projects are characterized by large collaborations – often international. Computing is an integral and inescapable part of High Energy Physics, which is a data and compute intensive science. Due to the need to push the boundaries of discovery and probe matter at the highest energies and intensities, HEP invents new technologies to enable the science. These include accelerator and detector technology, computational and data tools, as well as a system of distributed computing worldwide.

HEP values the High Performance resources available through NERSC, and the opportunity to help shape the choice of next generation of hardware through these NERSC Science Requirement Reviews. Allocation requests to HEP for NERSC use always exceed what is available and for 2013 requests are approximately double the availability. Traditionally the heaviest use of NERSC computing has been made by specific subsections of the HEP community who are well versed in the use of High Performance Computing (HPC) and in fact experts in some cases. These include the Lattice Gauge Theory community, Cosmic Frontier Simulations for specific experiments as well as those with wider reach, and accelerator modeling and simulation. Some of these users also represent the current HEP 'SciDAC' communities.

The NERSC Allocation process within HEP is becoming increasingly cognizant of the computing needs of all sections of the HEP research community. In part this is due to the awareness by the traditionally High Throughput Computing (HTC) communities that they too find NERSC resources valuable. Some of these groups – most notably the HEP theorists carrying out Monte Carlo Simulations that are used by experimentalists are beginning to use NERSC. HEP and ASCR have started a joint partnership effort to research into transforming the GEANT 4 code into one efficient to run on multi core and HPC platforms. In parallel, the Energy Frontier experimental community is preparing to avail of NERSC allocations, as are our two major Cosmic Frontier experiments. These 'new' user groups are expected to be very active users of NERSC for the time frame of this planning exercise.

HEP is an exciting program pushing ahead all three scientific frontiers. After decades of 'we have to find the Higgs Boson', we have now found 'A Higgs like Boson'. We have also found a long sought neutrino mixing angle. Our success has always been tied to advances in computing and other technology. As experiments become more precise and data volumes cross petabytes, HEP faces new computing, simulation, and data challenges. Current and future NERSC computers are expected to be a key resource for the HEP community.

For more information about HEP please visit http://science.energy.gov/hep.

3 About NERSC

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

The DOE Office of Science supports three major High Performance Computing Centers: NERSC and the Leadership Computing Facilities at Oak Ridge and Argonne National Laboratories. NERSC has the unique role of being solely responsible for providing HPC resources to all open scientific research areas sponsored by the Office of Science.

This report illustrates NERSC alignment with, and responsiveness to, DOE program office needs; in this case, the needs of the Office of High Energy Physics. 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. However, as demonstrated its users' scientific productivity, the combination of effectively managed resources, and excellent user support services the NERSC Center continues its 40-year history as a world leader in advancing computational science across a wide range of disciplines.

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

4 Meeting Background and Structure

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

In November 2012, the DOE Office of Advanced Scientific Computing Research (ASCR, which manages NERSC), the DOE Office of High Energy Physics (HEP), and NERSC held a review to gather HPC requirements for current and future science programs funded by HEP. This report is the result.

This document presents several findings, based upon a representative sample of projects conducting research supported by HEP. The case studies were chosen by the DOE Program Office Managers and NERSC staff to provide broad coverage in both established and incipient HEP research areas.

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

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

NERSC and ASCR have been conducting requirements workshops for each of the six DOE Office of Sciences offices that allocate time at NERSC (ASCR, BER, BES, FES, HEP, and NP). The process began in May 2009 and concluded in May 2011 for requirements with a target of 2014. A second round of meetings, of which this one was the second, began in September 2012 with a target for user needs in 2017.

Specific findings from the review follow**.**

5 Meeting Demographics

5.1 Participants

5.1.1 DOE / NERSC Participants and Organizers

Name	Institution	Area of Interest
Lali Chatterjee	DOE / HEP	HEP Program Manager
Sudip Dosanjh	NERSC	NERSC Director
Richard Gerber	NERSC	Review Facilitator
Dave Goodwin	DOE / ASCR	NERSC Program Manager
Barbara Helland	DOE / ASCR	Associate Director for ASCR (Acting), Director ASCR Facilities Division
James Siegrist	DOE / HEP	Associate Director for HEP
Harvey Wasserman	NERSC	Review Facilitator

5.1.2 Domain Scientists

Name	Institution	Area of Interest	NERSC Repo(s)
Julian Borrill	Lawrence Berkeley National Laboratory, UC Berkeley	Cosmic Background Radiation data analysis	Euclid, usplanck, mp107, planck, cosmosim
Richard Brower	Boston University	Lattice methods for QCD and statistical mechanics, quantum field theory of strings and particles	-
Andrew Connolly	University of Washington	Data management and analysis for LSST	m1727
Scott Dodelson	Fermi National Accelerator Laboratory	Dark Energy Survey Data Analysis	des
Cameron Geddes	Lawrence Berkeley National Laboratory	Laser-driven plasma wake field accelerators	m558
Steven Gottlieb	Indiana University	Lattice QCD	mp13
Salman Habib	Argonne National Laboratory	Cosmic Structure Probes of the Dark Universe	des, cusp, cosmosim, hacc

Stefan Hoeche	SLAC National Accelerator Laboratory	Particle physics phenomenology, in particular perturbative QCD and the construction of Monte Carlo event generators	m1738, m1758
Thomas LeCompte	Argonne National Laboratory	Physics Coordinator, ATLAS Experiment	m1092
Kwok Ko	SLAC National Accelerator Laboratory	Advanced Modeling for Particle Accelerators	m349
Peter Nugent	Lawrence Berkeley National Laboratory	Discovery and observation of supernovae, computational cosmology	m937, m1052, m779, m1276
Michele Papucci	Lawrence Berkeley National Laboratory	Supersymmety Studies at the LHC	m1610
Rob Roser	Fermi National Accelerator Laboratory	CDF Spokesperson, senior scientist and head of Scientific Computing Division at Fermilab	-
Elizabeth S Sexton-Kennedy	Fermi National Accelerator Laboratory	CMS experiment at the LHC, HEP data access and preservation	-
Panagiotis Spentzouris	Fermi National Accelerator Laboratory	Fermilab Accelerator and Detector Simulation and Support, PI of the SciDAC2 ComPASS project.	m1646
Doug Toussaint	University of Arizona	Lattice QCD	mp13, m1647
Frank Tsung	UCLA	Particle-driven plasma wake field accelerators	mp113
Craig Tull	Lawrence Berkeley National Laboratory	Scientific software frameworks, manager of software and computing for Daya Bay	dayabay
Torre Wenaus	Brookhaven National Laboratory	Physics Support and Computing Manager, U.S. ATLAS Operations Program	-

5.1.3 Observers

Name	Institution	Area of Interest
Lothar Bauerdick	Fermi National Accelerator Laboratory	CMS Center Director
Kenneth Bloom	University of Nebraska-Lincoln	U.S. CMS Tier-2 program leader
Jean Cottam	National Science Foundation	Computational and Data-Enabled Science and Engineering (CDS&E)
Peter Elmer	Princeton University	
Saul Gonzalez	National Science Foundation	Experimental Elementary Particle Physics
Randall Laviolette	DOE ASCR	Program Manager SciDAC

		Application Partnerships
Steven Lee	DOE ASCR	Program Manager
Lucy Nowell	DOE ASCR	Program Manager
Larry Price	DOE HEP	Program Manager Computational High Energy Physics
Nigel Sharp	National Science Foundation	Program Director for the LSST
Ceren Susut-Bennett	DOE ASCR	SC Program SAPs
Kathleen Turner	DOE HEP	Program Manager High Energy Physics
Mark Zisman	DOE HEP	General Accelerator R&D

5.2 NERSC Projects Represented by Case Studies

NERSC projects represented by case studies are listed in the table below, along with the number of NERSC hours each used in 2012. These projects accounted for about 85 percent of computer time used by HEP at NERSC that year.

NERSC Project ID (Repo)	**Project**	**Principal Investigator [presenter]**	**Hours Used at NERSC in 2012 (M)**	**Archival Data at NERSC 2012 (TB)**	**Shared Data on Disk (TB)**
lsst, boss, bigboss, des, dessn, ptf, desi, cosmo	*Experimental Cosmology*	Peter Nugent	2	40	20
cosmosim, cusp, hacc	*Cosmological Simulations for Sky Survey*	Salman Habib	24	70	120
planck, usplanck, mp107	*Cosmic Microwave Background Analysis*	Julian Borrill	13	550	200
m1400	*Supernova Studies*	Stan Woosley	13	100	3
mp13, m1647	*Lattice QCD*	Doug Toussaint, Steve Gottlieb	75	23	6
m778, m1646	*Community Petascale Project for Accelerator Science and Simulation (ComPASS)*	Panagiotis Spentzouris	3.8	30	10
m558	*Laser Driven Plasma Accelerator Simulations*	Cameron Geddes	12	160	2
mp113	*Plasma Based Accelerators*	Warren Mori [Frank Tsung]	8.3	90	0
m349	*Advanced Modeling for Particle Accelerators*	Kwok Ko	3.1	38	5
dayabay	*Intensity Frontier Data Analysis*	Craig Tull	1	214	500
Total of projects represented by case studies			**155**	**1,315**	**866**
NERSC 2012 HEP Total			**184**	**4,000**	
Percent Represented at Review			**84%**	**33%**	

6 Findings

6.1 Summary of Requirements

The following is a summary of consensus requirements for production computing, storage, and HPC services derived from the case studies.

6.1.1 Researchers need access to significantly more computational and storage resources to support HEP mission goals through 2017.

a) HEP research teams will need almost 43 billion hours of computing time in 2017, more than 200 times what they used in 2012 at NERSC.

b) Data storage needs are exploding as well, with a need for 225 PB of archival storage space in 2017, a factor of 57 times more than used at NERSC in 2012.

c) Access to these additional resources are critical enablers of high-profile scientific missions and facilities supported by HEP: e.g., the ATLAS and CMS experiments at the LHC; the LSST, DES, Planck cosmology projects and their follow-ons; the design of future accelerators; and a number of intensity frontier experiments like Daya Bay.

d) Progress in design, analysis, and control of systematic errors is already constrained by limited access to computing resources.

6.1.2 Scientists need vastly improved data I/O rates and better facilities for performing data-intensive science.

a) Without better I/O performance ever-larger simulations will not be able to output results and perform checkpoints in a reasonable amount of time, thus wasting simulation time while waiting on I/O. Time spent on I/O beyond 10 to 20 percent of the total run time is not acceptable.

b) Access times to archival storage must improve. Currently many research teams forgo use of cost- and energy-efficient tape storage technologies because of unacceptably long retrieval times.

c) Research teams need to efficiently process large data sets (both in volume of data and numbers of files), perform visualization and analysis on them, and share them among collaborators and the public. Researchers need databases and web portals.

d) The ability to archive and manage data is needed. Data sharing, curation, and provenance must be accommodated.

6.1.3 Research teams need to run both large-scale simulations and massive numbers of individual jobs.

a) Very large simulations – e.g., for cosmology or lattice QCD – are required to support interpretation of experimental results and test parameters of fundamental theories. These simulations will need to run millions of concurrent tasks/threads to complete their calculations.

b) High throughput computing (HTC); i.e., the ability to run massive numbers of jobs concurrently and/or quickly, is needed to support HEP data analysis, detector simulation design and response modeling, parameter studies, uncertainly quantification, and code verification and validation.

6.1.4 NERSC must help enable and ease the transition to next-generation architectures.

a) Since most codes will need to be rewritten or extensively modified to run efficiently on next-generation architectures, HEP research teams will need assistance transitioning their codes.

b) Access to early testbed machines is required for code development to prepare codes to run efficiently (or at all) on next-generation large production systems.

c) NERSC must supply training, support, and documentation to enable this transition.

6.2 Other Significant Observations

a) "Non-Traditional HPC" Communities: There are communities within DOE HEP that are not traditional users of large HPC centers that now have a profound need for computing, storage, and analysis facilities.

- ATLAS and CMS estimate a need for five times their 2012 usage of about 2.5 billion hours worldwide, or about 12 billion hours. They will need to store 190 PB of data.
- The ATLAS/CMS tier-1 and tier-2 compute usage of 2.5 billion hours in 2012 is equal to about twice the number hours NERSC delivered that year. In other words, worldwide computing for LHC HEP was the equivalent of about two NERSCs.
- This historic growth rate of HPC computing power (32X in five years) is greater than the projected growth rate of 5X need for ATLAS/CMS computing.
- There is a similar need for computing, storage, and analysis facilities for several research projects in the Cosmology Frontier, including LSST, DES, and BOSS.

b) GPU-based clusters are having a profound effect on a portion of the LQCD workload, the "analysis" phase. The increased capacity afforded by GPUs has resulted in changes in computational workflow and a significant increase in I/O demands in both intermediate and long term storage.

6.3 Computing and Storage Requirements

The following table lists the 2017 computational hours and archival storage needed at NERSC for research represented by the case studies in this report. "Total Scaled

Requirement" at the end of the table represents the hours needed by all 2012 HEP NERSC projects if increased by the same factor as that needed by the projects represented by the case studies.

6.3.1 Computing

Case Study Title	PI	Computing Needed in 2017	
		NERSC MPP Equivalent Hours (Millions)	Factor Increase
Experimental Cosmology	Nugent	82	41
Cosmological Simulations for Sky Survey	Habib	10,000	417
Cosmic Microwave Background Analysis	Borrill	500	38
Supernova Studies	Woosley	200	15
Lattice QCD	Toussaint, Gottlieb	24,000	320
Community Petascale Project for Accelerator Science and Simulation (ComPASS)	Spentzouris	85	22
Laser Driven Plasma Accelerator Simulations	Geddes	1,000	83
Plasma Based Accelerators	Mori	166	20
Advanced Modeling for Particle Accelerators	Ko	5	1.6
Intensity Frontier Data Analysis	Tull	8	8
Perturbative QCD and Phenomenology	Hoeche	15	-
Total Represented by Case Studies		**36,000**	
% of NERSC HEP Represented by Case Studies		**84%**	
All HEP at NERSC Total Scaled Requirement 2017		**42,735**	**232**
Energy Frontier Data Analysis (Worldwide)		75,000	50

6.3.2 Storage

Case Study Title	PI	Archival Data Storage Needed in 2017		Shared Online Data Storage Needed in 2017	
		TB	Factor Increase	TB	Factor Increase
Experimental Cosmology	Nugent	1,000	25	500	25
Cosmological Simulations for Sky Surveys	Habib	10,000	143	10,000	83
Cosmic Microwave Background Analysis	Borrill	50,000	91	5,000	25
Supernova Studies	Woosley	2,000	20	200	67
Lattice QCD	Toussaint, Gottlieb	200	8.7	20	3.2
Community Petascale Project for Accelerator Science and Simulation (ComPASS)	Spentzouris	500	17	100	10
Laser Driven Plasma Accelerator Simulations	Geddes	5,000	31	600	300
Plasma Based Accelerators	Mori	1,800	20	0	0
Advanced Modeling for Particle Accelerators	Ko	50	1.3	20	4
Intensity Frontier Data Analysis	Tull	4,000	19	2,000	4
Perturbative QCD and Phenomenology	Hoeche	5,000	-	200	-
Total Represented by Case Studies		**74,500**		**18,640**	
% of NERSC HEP Represented by Case Studies		**33%**		**unknown**[1]	
All HEP at NERSC Total Scaled Requirement 2017		**227,000**	**57**	**unknown**[1]	**21.5**
Energy Frontier Data Analysis (Worldwide)		190,000	2.5		

[1] Aggregate HEP shared usage in the /project file system was not available.

7 NERSC Plans and Initiatives

Pertinent to the summary requirements given above NERSC has a number of initiatives and plans underway. They are briefly mentioned here.

7.1 Computational and Storage Resources

Through 2011 NERSC delivered to the DOE science community a Moore's Law-like two-fold year-to-year increase in computational hours. Due to a combination of funding and technological constraints, 2012 usage was about a factor of two less than that trend would have predicted. The acquisition of the Edison system in 2013 will increase 2012 allocations by about a factor of 2.5 and the target for the NERSC-8 system, scheduled to be in operation in 2016, will put NERSC back on, or near, the historical trend, depending on actual realized funding. See (Figure 1).

Meanwhile, the needs of the HEP community continue to grow. In the first HEP review from 2009, researchers estimated a need for 2.4 billion hours at NERSC in 2014. Given that HEP currently receives about 15 percent of the NERSC total allocation, HEP projects will be awarded on the order of 450 million hours for 2014, less than 20 percent of their stated need. By 2017, HEP researchers estimate needing more HPC computing cycles (42.5 billion) than will be provided by the entire NERSC-8 system under current budget scenarios.

NERSC expects to continue to grow its archival storage capacity at historical rates (Figure 2). As with computing, the HEP requirement for 2017 is expected to far above the trend line.

7.2 I/O Rates and Data-Intensive Science

NERSC continues to scale I/O subsystems to meet the demands of ever-larger simulations. The Edison system, which will be in production in 2014, features a scratch file system with more than twice the bandwidth (>140 GB/sec) of the Hopper system and more than three times the storage space (6.4 PB vs. 2 PB).

NERSC continues to be a leader in data-driven web science portals (see http://portal.nersc.gov/) and is working to provide a scalable infrastructure that will support science teams' needs for building interfaces to access and share data. Work toward deploying scalable parallel databases for science is also underway.

7.3 HPC and HTC Workloads

Jobs run at NERSC use both large-scale HPC as well as HTC workflows and NERSC recognizes both are important for scientific productivity. NERSC has supported HTC through various activities (with, for example, the Joint Genome Institute, the Materials Project, the Daya Bay neutrino experiment, ATLAS, and ALICE) and will continue to help integrate HTC onto leading-edge NERSC systems where appropriate through workflow tools, "task farmers," and "compatibility" software like Cray's Cluster Compatibility (CCM) mode.

7.4 Application Readiness

NERSC already has an "application readiness" team that will be working with a small number (about 10) of codes teams to prepare applications for the next generation of HPC systems. Lessons learned and best practices will be incorporated into the NERSC web site and will form the basis for NERSC training efforts in preparation for the NERSC-8 system. NERSC also plans to provide testbed systems for users when further details about the next large system are available.

7.5 Non-Traditional Workloads

The computing and store requirements of "non-traditional" workloads (e.g., LHC data analysis, Intensity Frontier experiments) exceed the capacity of their secured resources over the next 5-10 years. NERSC, with its past experience in this area and in its unique role of supporting all SC research, is a natural potential source of resources for these workloads. Making resources available to these communities would have to come from the HEP NERSC program managers through the normal NERSC ERCAP allocation process or from an additional funding stream.

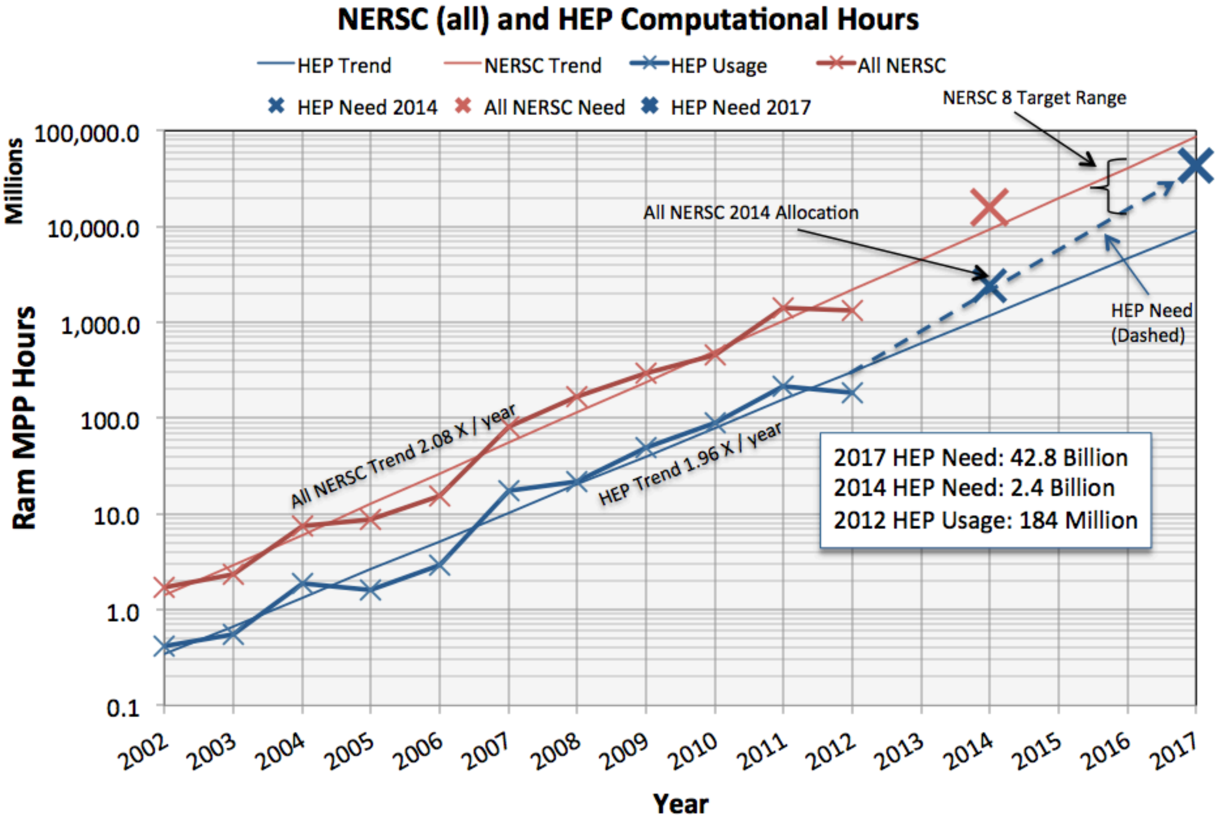


Figure 1 Historical usage (solid lines with markers) and trend lines for NERSC usage in MPP Hours, hours normalized to one Hopper core-hour.

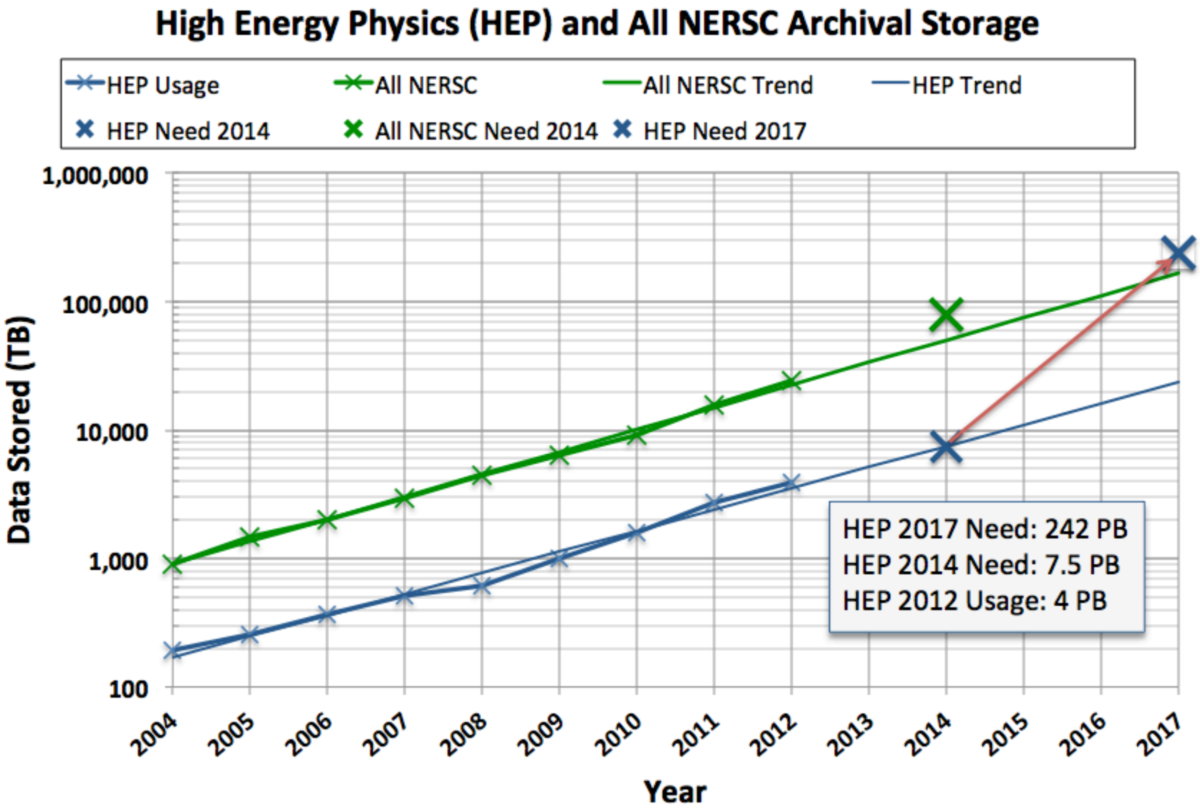


Figure 2 All NERSC (green line) and HEP data (blue) stored on NERSC's HPSS archival storage system. The crosses are usage estimated from the requirement reviews. The red line shows that the anticipated growth exceeds the historical growth rate.

8 Cosmic Frontier Case Studies

The following four case studies are representative of major research efforts in the Cosmic Frontier over the next five years.

8.1 Cosmological Simulations for Sky Surveys

Principal Investigator: Salman Habib (ANL)
Key Investigators and Institutional PIs:
Jim Ahrens (LANL), Ann Almgren (LBNL), Scott Dodelson (FNAL), Nick Gnedin (FNAL), Katrin Heitmann (ANL), David Higdon (LANL), Peter Nugent (LBNL), Rob Ross (ANL), Anze Slosar (BNL), Risa Wechsler (SLAC), Martin White (UC Berkeley)

NERSC Repositories:
1) Cosmological Simulations for Sky Surveys (cosmosim)
2) SciDAC-3: Computation-Driven Discovery for the Dark Universe (cusp)
3) Dark Energy Survey (des, not covered here)
4) Analysis and Serving of Data from Large-Scale Cosmological Simulations (hacc, NERSC Data Pilot project)

8.1.1 Project Description

8.1.1.1 Overview and Context

The current science thrusts of DOE HEP's Cosmic Frontier program are dark matter and dark energy, high energy cosmic and gamma rays, and studies of the cosmic microwave background (CMB). As a core aspect of the Cosmic Frontier, modern cosmology is one of the most exciting areas in all of physical science. Progress over the last two decades has resulted in cementing a 'Consensus Cosmology' that, defined by only half a dozen parameters, is in excellent agreement with a host of cross-validated observations. Although the fact that a simple model can be so successful is already remarkable, three of its key ingredients – dark energy, dark matter, and inflation – point to future breakthrough discoveries in fundamental physics, as all require ingredients beyond the Standard Model of particle physics.

As drivers of progress along the broad front of the Cosmic Frontier research program, large-scale computing and data storage are of central importance. Computational cosmology functions in three key roles:

1) Providing the direct means for cosmological discoveries that require a strong connection between theory and observations ('precision cosmology');
2) As an essential 'tool of discovery' in dealing with large datasets generated by complex instruments; and,
3) As a source of high-fidelity simulations that are necessary to understand and control systematics, especially astrophysical systematics.

High-end computing and storage are key components of cosmological simulations, which are among the largest computations being carried out today. These can be classified into two types: gravity-only N-body simulations, and 'hydrodynamic' simulations that incorporate gas dynamics, sub-grid modeling, and feedback effects. Because gravity dominates on large scales, and dark matter outweighs baryons by roughly a factor of five, gravity-only N-body simulations provide the bedrock on which all other techniques rest. These simulations can accurately describe matter clustering well out into the nonlinear regime, possess a wide dynamic range (Gpc to kpc, allowing coverage of survey-size volumes), have no free parameters, and can reach sub-percent accuracies. Several post-processing strategies exist to incorporate additional physics on top of the basic N-body simulation. Whenever the dynamics of baryons is important, substantially more complex computations are required. 'Gastrophysics' is added via either grid-based adaptive mesh refinement (AMR) solvers or via particle-based methods such as smoothed-particle hydrodynamics (SPH). Both classes of simulations require cutting-edge resources, and face limits imposed by the size and performance of even the largest and fastest supercomputers. Because large datasets with ever increasing complexity are routinely created by these simulations, major storage and post-processing requirements are associated with them. These are already at the ~PB level, and are bound to increase steeply with time.

8.1.1.2 Scientific Objectives for 2017

The global "project," which consists of three repositories at NERSC, and simulation and storage allocations at other centers (via ALCC, INCITE, and other awards), is targeted at multiple cosmological probes that are connected to the data stream from large-scale sky surveys, both ongoing and planned for the future. Ongoing surveys include the Baryon Oscillation Spectroscopic Survey (BOSS), the Dark Energy Survey (DES), and the South Pole Telescope (SPT). Future surveys include the Large Synoptic Survey Telescope (LSST) and the Mid-Scale Dark Energy Spectroscopic Instrument (MS-DESI) spectroscopic survey. To a very large extent, the simulation program is tied to the discovery science potential of these surveys. The different cosmic probes are each associated with a major computational campaign, of varying specificity. Below, we list a subset of important probes and associated computational projects that provide a flavor of the challenges ahead, leading on to the specific workplan that would eventually be implemented in 2017.

Baryon acoustic oscillations (BAO) accessed from galaxy surveys [BOSS, DES, MS-DESI, LSST] together provide a precision measurement of the geometry of the Universe at $z < 1.6$. The challenge here is the ability to run large-volume N-body simulations that can precisely determine the BAO signature in the power spectrum or the corresponding peak structure in the correlation function. At $z > 2$, the BAO signature can be extracted from the spatial statistics of the quasar Ly-α forest (BOSS, MS-DESI) – a probe of the intervening intergalactic medium (IGM). This requires running large hydrodynamics simulations to model the distribution of neutral hydrogen. *Cluster counts* (DES, LSST) provide measurements of both geometry and structure growth. Here, large-volume N-body simulations are required to provide sufficient statistics, and hydrodynamic simulations are necessary to characterize observable-mass relations. *Weak gravitational* *lensing* (DES, LSST) has multiple uses – measurements of geometry, structure growth, and cluster masses. All of these need large N-body and hydrodynamic simulations to accurately predict the mass distribution responsible for the lensing signal. *Redshift-space distortions* (BOSS, DES, LSST, MS-DESI) measure the growth of structure and can test theories of modified gravity; these require large-volume N-body simulations to determine and characterize individual galaxy velocities. *Ly-α forest* measurements of the matter power spectrum (BOSS, MS-DESI) are sensitive to small length scales and hence to probing the neutrino mass and thermal weakly interacting massive particle (WIMP) mass limits. Properly exploiting this probe demands hydrodynamics simulations with radiative transfer to interpret quasar spectra.

A significant computational task relates to the analysis of large datasets sourced by sky surveys and by simulations. The observational datasets are expected to range from ~1 PB for DES and ~100 PB for LSST, while simulation data generation is constrained only by storage and I/O bandwidth and can potentially produce much larger datasets. Traditional high performance computing platforms are quite unsuited to data-centric computations, and new approaches to scalable data-intensive computing are needed, certainly by 2017, if not much sooner. It is also apparent that managing a complex workflow with very large datasets will be a significant component of computing at the cosmic frontier. Aside from the intrinsic difficulties in theoretical modeling of the individual and collective science cases and dealing with large observational datasets, there is a major added complication: Data analysis in cosmology is in fact a high-dimensional problem of statistical inference where one solves for cosmological and modeling parameters, requiring many solutions of the forward model (predictions for the observations) within a Markov chain Monte Carlo (MCMC) framework. A large number of (de-rated) simulation runs are also needed to determine error covariances. These requirements motivate the development of a new set of fast statistical techniques that at the same time can provide results with small, controlled errors. These sorts of techniques will have become ubiquitous by 2017.

8.1.2 Computational Strategies (now and in 2017)

8.1.2.1 Approach

Computational cosmology is not by any means a single computational problem, but rather an interconnected and complex task combining forward predictions, observations, and scientific inference, all within the arena of high performance computing and very large datasets. The primary concern here is with structure formation-based probes; all of these essentially measure – directly or indirectly – the dark matter-dominated density field, or quantities related to it. The success of the overall approach rests on a solid first principles understanding of the basis of structure formation: very close to Gaussian initial fluctuations laid down by inflation (or some other process), to be later amplified by the gravitational instability giving rise to the complex structures observed today (the growth rate of structure is a competition between the attraction of gravity and the expansion of space). In a standard cosmological analysis, this process is fully described by general relativity and atomic physics.

The central computational problem is to generate accurate initial conditions (multi-scale, multi-species, as needed) and then to solve the Vlasov-Poisson equation for the purely gravitating species (dark matter) and include, along with gravity, gasdynamics, feedback, radiative processes, and sub-grid models for the baryonic matter. Because of the conflicting requirements of simulation volume and detailed treatment of small-scale physics, gravity-only N-body codes are used to handle the larger volumes, whereas hydro simulations are run at smaller volumes. One significant simulation task is to build a picture that can consistently include the information from hydro simulations within N-body runs. The data generated by the simulations can be very large and the global analysis task (e.g., constructing synthetic sky catalogs) is itself as complex as carrying out the simulation runs. Thus building the analysis frameworks is also a significant aspect of the overall computational strategy.

Finally, extracting science by combining simulation results and observational data is a separate endeavor, requiring the use of an MCMC framework (and alternatives), where forward predictions must be generated tens to hundreds of thousands of times. The complexity of a single prediction, both in terms of physics and numbers of parameters, obviously precludes brute force simulation runs as a viable approach. To overcome this problem, the 'Cosmic Calibration Framework' has been recently developed. The main idea behind the framework is to cover the cosmological and astrophysical model space in an efficient manner by using sophisticated statistical sampling methods and techniques for functional interpolation over high-dimensional spaces. In addition, because the cosmological 'response surface' is relatively smooth, and the current observational constraints limit the prior range substantially, the number of required simulations can be brought down into the hundreds, and allows MCMC analyses to be carried out with very fast numerical oracles for cosmic probes, the so-called emulators. Simulation campaigns to generate these emulators will be a key component of the planned work for 2017.

8.1.2.2 Codes and Algorithms

Large-scale cosmological N-body codes are essential for the success of all future cosmological surveys. As supercomputer architectures evolve in more challenging directions, it is essential to develop a powerful next generation of these codes that can simultaneously avail various types of many-core and heterogeneous architectures. This is the driver behind the development of the HACC (Hardware/Hybrid Accelerated Cosmology Codes) framework. HACC's multi-algorithmic structure also attacks several weaknesses of conventional particle codes including limited vectorization, indirection, complex data structures, lack of threading, and short interaction lists. It combines MPI with a variety of local programming models (e.g., OpenCL, OpenMP) to readily adapt to different platforms. Currently, HACC is implemented on conventional and Cell/GPU-accelerated clusters, on the Blue Gene architecture, and is running on prototype Intel MIC hardware. HACC is the first, and currently the only large-scale cosmology code suite worldwide that can run at scale (and beyond) on all available supercomputer architectures.

HACC uses a hybrid parallel algorithmic structure, splitting the gravitational force calculation into a specially designed grid-based long/medium range spectral particle-mesh (PM) solver (based on a new high performance parallel 3D FFT, high-order Greens function, super-Lanczos derivatives, and k-space filtering) that is common to all architectures, and an architecture-tunable particle-based short/close-range solver. The grid is responsible for four orders of magnitude of dynamic range, while the particle methods – a blend of direct particle-particle and recursive coordinate bisection (RCB) tree/fast multipole (implemented via the pseudo-particle method) algorithms – handle the critical two orders of magnitude at the shortest scales where particle clustering is maximal and the bulk of the time-stepping computation takes place. Using a benchmark run of 3.6 trillion particles, HACC has demonstrated outstanding performance at close to 14 PFlops/s on the BG/Q (69% of peak) using more than 1.5 million cores and MPI ranks, at a concurrency level of 6.3 million. This is the highest level of performance yet attained by a science code on any computer. Production runs on Hopper with 30 and 68 billion particles have recently been carried out and test runs on Edison will begin soon. HACC development continues in several directions (optimization for Titan, I/O optimization, load balancing improvements, mapping to new architectures). An integrated *in situ* analysis framework, essential to reduce the I/O and storage workloads, is another HACC feature.

Aside from HACC, we also use Gadget, a public domain code developed primarily by Volker Springel, and TreePM, a similar code developed primarily by Martin White. Both codes use the TreePM algorithm and can scale to ~100K cores in MPI/OpenMP mode.

The collaboration brings together two state of the art cosmological hydrodynamics codes, ART and Nyx. The aim is to use and develop them in synergistic ways, so that they are applied to suit their respective strengths, yet with enough overlap such that results can always be tested with more than one code. The Adaptive Refinement Tree (ART) code is a high-performance cosmology code originally developed in the mid-nineties. Hydrodynamics capabilities were added later, and the code now includes several additional aspects of the physics relevant for the formation of galaxies and clusters. Examples of recent results from ART include an early study of the baryonic effects on weak lensing measurements and a new low-scatter X-ray mass indicator for galaxy clusters. Nyx is a newly developed N-body and gas dynamics code designed to run large problems on tens of thousands of processors. It is based on the BoxLib framework for structured grid adaptive mesh methods, supported as part of the SciDAC FASTMath Institute, and underlies a number of DOE codes in astrophysics and other areas. The use of BoxLib enables Nyx to capitalize on extensive previous efforts for attaining high performance on many processors. The parallelization strategy uses a hierarchical programming approach; excellent weak scaling of the hydrodynamic framework has been demonstrated up to 200K processors. Nyx has been successfully tested using two cosmology code comparison suites.

Both ART and Nyx follow the evolution of dark matter particles gravitationally coupled to a gas using a combination of multi-level particle-mesh and shock-capturing Eulerian methods. High dynamic range is achieved by applying adaptive mesh refinement to both gas dynamics and gravity calculations. The parallelization strategies implemented in ART and Nyx employ both MPI and OpenMP. Multigrid is used to solve the Poisson equation for self-gravity. In both codes, the same mesh structure that is used to update fluid quantities is also used to evolve the particles via the particle-mesh method.

However, the two codes differ fundamentally in their approach to adaptivity. ART performs refinements locally on individual cells, and cells are organized in refinement trees, whereas Nyx uses a nested hierarchy of rectangular grids with refinements of the grids in space.

The ART code allows for modeling a wide range of physical processes. Specifically, the current version of the code includes the following physical ingredients (in addition to gravity, dark matter, and gas dynamics): (i) detailed atomic physics of the cosmic plasma, including a novel method for modeling radiative cooling of the gas; gas cooling functions implemented in ART are the most accurate of all existing cosmological codes; (ii) the effects of cosmic radiation on the gas; and (iii) formation of stars and their feedback on the cosmic gas. All this functionality is directly relevant to our science goals. Nyx contains the 'stubs' for attaching these types of additional physics packages.

8.1.3 HPC Resources Used Today

8.1.3.1 Computational Hours

The three allocations combined used 24M compute hours in 2012 at NERSC. One of them (cusp repository) is primarily for larger-scale SciDAC runs, The cosmosim project covers a set of more general simulations and analysis projects, including code tests, while hacc is a special pilot project for data-intensive computing. The 2013 allocation is distributed as follows: cusp, 16M compute-hours, cosmosim, 5M compute-hours, hacc, 5M compute-hours. The HACC team has a large allocation as one of the ALCF Mira Early Science Projects (150M compute-hours in 2013), have been running on Mira (IBM BG/Q) while it is under acceptance, and also on Intrepid (IBM BG/P) as part of HACC framework development. The team also has an ALCC project at ALCF and OLCF on Mira and Titan for 2012/2013 with 10M compute-hours and an INCITE award at ALCF for 2013 with 40M compute-hours. In addition, the entire collaboration has access to a number of local clusters including CPU/GPU resources (such as Dirac at NERSC). In general, considering the computational capabilities we have, and the diversity of science needs, the overall project is currently allocation-starved and storage-limited.

8.1.3.2 Compute Cores

The number of cores varies widely depending on the science problem, ranging from ~2K to 65K for production runs on Hopper. Production runs have been carried out with ART, Gadget, and HACC, as well as initial runs with Nyx. We could certainly use more cores given our science case, but our total allocation is quite limited. For instance, a single, 68-billion particle run for BOSS covariance predictions with HACC takes less than 1M compute-hours (on 65K cores) but about 60 such simulations are actually needed, which is significantly larger than the current allocation. For HACC there is no maximum number to the core count, and we could use all of Hopper if our allocation allowed it. We note that throughput on the 65K run was very good, so simulation campaigns at this scale or larger on Hopper are very viable. Extrapolating from this, Edison will likely be a very valuable resource.

The AMR codes run into memory and scaling limitations; ART currently scales to tens of thousands of cores, while Nyx, in principle, can scale to the full machine, but to run some of the large problems of interest, more memory would be required. Nyx has been run primarily in testing and validation mode so far. Multiple ART production runs have been carried out on Hopper using ~5-10K cores each. We do have multiple jobs running concurrently with the different codes we use, but usually no more than a few production runs at a time.

8.1.3.3 Shared Data

We have three project directories – cosmosim, cusp, and hacc – containing about 120 TB of data stored in over 1 million files. Of that total, about 110 TB is associated with the hacc data intensive pilot.

8.1.3.4 Archival Data Storage

In SRU units, we are not using HPSS much (70 TB, cosmosim – 28%, cusp – 5%, hacc – 2%), generally because of the slow read times for large files. Our experience shows that it may be faster to move data to a remote file storage system and then transfer to NERSC via GlobusOnline than to use HPSS. We have received assistance from NERSC in this area and hope to be able to improve our usage of the HPSS resource.

8.1.4 HPC Requirements in 2017

8.1.4.1 Computational Hours Needed

The targeted number of compute-hours is difficult to estimate reliably so far in advance. Based on our current usage and projecting the science demand and systems availability, the number of compute hours in Hopper units will be in the range of ~3-10 billion compute-hours. If we use the rough factor of a 30X increase over 2012 allocation levels as suggested by historical trends it would imply an allocation of 600M compute-hours in 2017 at NERSC, significantly less than that desired. Other resources we will use in 2017 include systems at the ALCF and OLCF – which will have upgraded their systems on this timescale to be more than another order of magnitude faster and larger than Mira and Titan. Access to NSF resources at NCSA and XSEDE systems is another possibility.

8.1.4.2 Number of Compute Cores

The possibility of having conventional compute cores in large systems in 2017 is quite remote, so our codes are not really targeted to evolve in this direction. Nevertheless, the number of lightweight cores in large systems is likely to be in the 10-100M range, although the number of MPI ranks will be significantly smaller, because of the small memory footprint per core. For production runs, we would estimate running on 100K to 10M cores, with the number of MPI ranks being a factor of ten smaller. HACC has already demonstrated running up to 1.5M cores and the same number of MPI ranks, with 4 OpenMP threads per core. Depending on the application, ART will scale to ~200K to 1M cores by 2017. Nyx will follow the development of BoxLib along an exascale trajectory, and given its current scaling to ~100K cores, it is expected that even the current approach will scale to 1M cores. It will be difficult to imagine the AMR codes running at less than 1 GB per MPI rank. As with the current use case, we do expect to be running multiple jobs concurrently, but the number of such jobs is likely to be small, since the footprint of each job will be substantial.

8.1.4.3 Data and I/O

The size of the data to be read in and written out is largely determined by the checkpointing strategy. If we assume a similar strategy as used currently, i.e., dumps of the entire system state, then these individual files can be quite large, of the order of 100 TB for production run checkpoints. However, if local NVRAM is available, then large checkpoints can be avoided. Because of large file sizes, and the associated I/O and storage limitations, most of our production codes will likely move over more to an in situ analysis based approach, which will reduce the amount of data stored considerably. We also plan to use both lossless and lossy compression strategies. By 2017, storage requirements will likely hit 10s of PB, but we would like a way to analyze the stored data that does not involve going back to the computer that generated it. It would be useful to have a scalable data-intensive computational resource for this task (this could be integrated within the storage system, but it is not clear that such a system can be ready by 2017).

The I/O bandwidth requirements to the file system will also be set by checkpointing constraints and the mean time between failures (MTBF) figure for the machine. If a 100 TB file has to be dumped in, say, 10 minutes, then this corresponds to an I/O bandwidth of 150-200 GB/s, which is available today. On the other hand, if the machine is relatively unstable, then one may wish to improve this performance (or if the occasional very large job requires a much higher number). We would tolerate something like 10-15% of total runtime to checkpoint I/O. The other I/O requirements are science-driven, and not as severe, so we could probably tolerate another 10% for those. It should be noted that a stable I/O system is necessary for these estimates to be valid. Currently, we see a fair number of code crashes caused by I/O system failures at checkpointing attempts.

8.1.4.4 Shared Data

We would like to have something available in the ~10 PB class (which is consistent with the hacc allocation of 300 TB times a factor of 30). As an example, a current trillion-particle run on Mira has already generated 6 PB of data; a distilled version of this will be moved over to NERSC in the near future. The caveat is, however, that much of it may not be useful for re-analysis if the only machine available for this is the host supercomputer. Some thought should be devoted to this – whether to add conventional cluster resources or a data-intensive compute resource designed more along the lines of ‘active storage’ with ‘Infrastructure as a Service’ capabilities.

8.1.4.5 Archival Data Storage

Although not everyone in our collaboration may agree with this particular point of view, for the most part, raw archival simulation data has limited use. (For example, the history of HPSS use is relatively sparse.) Such data is hard to get at and only rarely is a dataset so valuable that it needs to be archived for safety's sake, although this may change somewhat with time. In the near future, it is unlikely that one would need to archive data at a scale any bigger than the file system storage. So, once again, this would imply something in the ~10 PB class. (However, as storage costs continue to fall, spinning disk can replace tape, and this use case may need to be rethought.)

8.1.4.6 Memory Required

The memory requirements are simply a function of the problem run size. For a production N-body simulation in the trillion-particle class, this is roughly 100 TB, when one includes grid memory and analysis overheads. AMR codes typically have much larger overheads so by 2017, one could imagine large N-body runs (multi-trillion particles) in the 0.5 PB range, and AMR runs at base grid sizes up to $10,000^3$ in the multi-PB range. These numbers are not so far from what is the maximal used currently, for example, we ran HACC on the entire Sequoia system (1.5 PB RAM) at >40% memory usage.

The node-level memory requirements vary considerably across N-body and AMR codes. Whereas, HACC can tolerate node-level or MPI rank-level memory of 1 GB, realistic cosmology simulations with AMR will require significant effective memory per core, although the exact amount would depend on OpenMP (or equivalent) performance – it is unlikely that such codes would use less than 1 GB per effective compute core. In the case of ART, for example, it will be important in the future to achieve good OpenMP performance across several chips with memory-on-chip systems, thereby greatly reducing the memory-per-core requirement. However, it is not clear whether such systems will be available in 2017. A better metric than memory-per-core is memory-per-task, which probably will not change much and will always remain in the range of 10-15 GB. The achievable OpenMP performance will then determine the memory-per-core value.

To summarize, the memory per NUMA node should remain as large as is practicable, certainly more than 16 GB would be a good baseline value although 8 GB might be acceptable. (As you can imagine, all applications teams will ask for as much memory per core as you can get us!) But the current value for the Xeon Phi is probably already getting uncomfortably close to the minimum that most applications can tolerate without code rewrites, of roughly ~100 MB/core. The success or failure of the OpenACC approach will certainly bear on this.

8.1.4.7 Many-Core and/or GPU Architectures

HACC is fully ready for both many-core and GPU accelerated systems; it was originally designed and run on Roadrunner, the first large accelerated system, and since then has been run on CPU/GPU clusters and at full scale on Titan, with very good performance being achieved. HACC has also been run on MIC prototypes. How HACC handles these different modes of operation is by separating the long-range force computation and its communication strategy from the short-range force solvers, which are modular and can be plugged in and out of the overall HACC framework. The short-range solvers use different algorithms (direct particle-particle, local and hyper-local trees), programming models, and data movement strategies, depending on the local architecture of the system (GPU or MIC-accelerated, for example). The same strategy is used to optimize for non-accelerated systems, for instance the short-range solver for the BG/Q is optimized differently than the one for a Cray XE6.

For AMR codes, it is probably fair to say that efficient implementations on next-generation architectures are still a research problem. Computationally expensive pieces such as atomic physics can be sent to the accelerator, but it is unlikely to improve the overall performance until the radiative transfer is also moved over to the GPU. For many-core systems, the small memory per core, and the cost of nonlocal memory access within and across nodes is a major hurdle for the block-structured approach. Considering the complexity of the overall task, it is improbable that our AMR codes – both ART and Nyx – will be fully ready for many-core and GPU-acceleration by 2017, unless significant manpower is devoted to this task.

It is possible that vendor-supplied pragma-based approaches could function as a stopgap measure when transitioning to accelerated and many-core systems (e.g., OpenACC), but it is unlikely that the performance gains from these would be significant, as has been demonstrated by several porting attempts to large CPU/GPU systems, unless a fair amount of work is put in to restructure the codes. Tolerating lower performance, but just getting a code to run on these more complex architectures may be an inevitable part of transitioning to them.

A key issue here is code portability. For a complex code, it will be unlikely that it will be worth the effort to restructure it so that it runs well on a single architecture. Hence, the directive-based approach may be a useful compromise for this reason as well. It is also unrealistic to assume that a DSL-based solution will exist on the 2017 timeframe as anything but a research project.

NERSC could help with this task by working with some early science teams to test out various possibilities and gain experience in what works (and what does not) with directive-based systems, especially the current OpenACC, since, for example, PGI accelerated compilers will support both GPUs and the Xeon Phi.

8.1.4.8 Software Applications and Tools

It is unlikely that our effort will need anything too different from the current software base, which is relatively minimal (MPI, OpenMP, C, C++, F90, FFTW, viz packages, etc.). Within HACC, we have our own I/O acceleration system based on the Glean framework, and we are exploring PnetCDF as a file format (performance with PnetCDF has been tested on Hopper with good results). ART and Nyx have their own parallel I/O systems.

8.1.4.9 HPC Services

Current NERSC services are extremely useful. In our opinion, NERSC remains the easiest supercomputing center to work with for an external user. Certainly, services such as consulting or account support, data analytics and visualization, training, support servers, collaboration tools, web interfaces, federated authentication services, gateways, etc. should all exist. It may be a good idea to focus on training the NERSC community to move over to new architectures sooner rather than later. (This was very helpful to many people when the switch from vectors to MPPs was made at NERSC in the mid-90s.)

8.1.4.10 Time to Solution and Throughput

We will not have special needs in this regard. More flexible job scheduling may be something for NERSC to explore, but it is not clear what the boundary conditions are for this.

8.1.4.11 Data Intensive Needs

As already indicated above, cosmology codes create large amounts of data that must be analyzed, and the results of these analyses in turn produce next-level data as well as object catalogs. Handling this creates the need for a number of services and tools that NERSC should help to provide. This should be a logical extension of current practice; the pilot project for data-intensive computing has been a very good way for us to get started in constructing a portal-based analysis service for cosmological simulation data. We are very satisfied with the progress on this project and expect to launch the service publicly this summer.

8.1.4.12 Requirements Summary Worksheet

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours	24 M	~3-10 G
Typical number of cores* used for production runs	2K-64K	100K-10M

Maximum number of cores* that can be used for production runs	HACC – no limit, ART – 20K, Nyx – 100K	HACC – no limit, ART/Nyx–up to 1M
Data read and written per run	Up to 10s of TB	Up to 100's of TB
Maximum I/O bandwidth	3-10 GB/sec	150-300 GB/sec
Percent of runtime for I/O	Max ~25% (includes checkpoints)	Max ~25% (includes checkpoints)
Shared filesystem space	120 TB (400 TB very soon)	~10 PB
Archival data	70 TB	~10 PB
Memory per node	24 GB	8 GB
Aggregate memory	~10 TB (only because large runs will burn up allocation)	Up to 0.5 PB

*Traditional cores

8.2 Experimental Cosmology

Worksheet Authors: Andrew Connolly (University of Washington), Peter Nugent (Lawrence Berkeley National Laboratory)
NERSC Repositories: lsst, boss, bigboss, des, dessn, ptf, desi, cosmo

8.2.1 Project Description

8.2.1.1 Overview and Context

This decade will witness a dramatic increase in the need for computation and the amount of data coming from a new generation of cosmological experiments (i.e., large-scale imaging and spectroscopic surveys). The objective of these experiments is to understand the nature of dark energy and dark matter, one of the most fundamental unknowns in physics today, impacting our understanding of particle physics, cosmology, and possibly theories of gravity. Experimental cosmology addresses these questions through the use of multiple, complementary observational probes: gravitational weak lensing to study the growth of structure and geometry, baryon acoustic oscillations to measure the angular-diameter distance vs. redshift relation and Type Ia supernovae to measure the luminosity distance vs. redshift relation.

Current surveys – the Baryon Oscillation Spectroscopic Survey (BOSS), the Palomar Transit Factory (PTF) and the Dark Energy Survey (DES) – are mapping thousands of square degrees of the sky over a period of several years. The next generation of experiments – BigBOSS, the Zwicky Transit Factory (ZTF), and the Large Synoptic Sky Survey (LSST) – will increase the data generation rate by an order of magnitude.

This increase in survey data means that statistical noise will no longer determine the accuracy to which we measure cosmological parameters. The control and correction of systematic uncertainties will determine the scientific impact of any cosmological survey. Achieving the goals of current and planned experiments will, therefore, require the processing and analysis of experimental data streams, the development of techniques and algorithms for identifying cosmological signatures and for mitigating systematic uncertainties (thereby optimizing the science of interest to the Office of High Energy Physics), and detailed cosmological simulations for use in interpreting systematic uncertainties within these data. (Cosmological simulations to support this work are detailed in “Cosmological Simulations for Sky Surveys” case study in this report.)

The computational challenges may be divided into three principal areas of research: n-body and hydrodynamic simulations of the formation and evolution of structure within the universe; photon-based simulations of the data expected from planned surveys; and analysis pipelines for processing the petabyte data sets expected to be generated by the next generation surveys. Of these, the n-body and hydro simulations are treated elsewhere in this report (See Cosmological Simulations for Sky Surveys) and we consider here only the latter two.

8.2.1.2 Scientific Objectives for 2017

In 2017 the Dark Energy Survey (DES) is expected to be entering its final year of operations. This experiment will have surveyed 5000 square degrees of the southern sky in 5 optical filters from the ultraviolet to near infrared. It will have detected 300 million galaxies and several thousand supernovae using a 57 Megapixel camera on the Blanco 4m telescope in northern Chile. On completion the DES will be the largest cosmological survey with an archive close to 4 PB in size. At brighter magnitude limits, the Zwicky Transit Factory is expected to begin operations in 2015. Designed to identify transient sources it will generate 10 GB per minute with an archive of >1PB of data per year and a total of ~5x10^{10} detected sources.

On longer timescales, the LSST expects to achieve first light in 2019 and be fully operational in 2021. By 2017 the processing pipelines are required to be capable of analyzing simulated LSST data at 10% of the operational data rate (i.e. 2TB of data in 24 hrs). Processing of these data will include: measuring the shapes of galaxies by combining repeated observations of the same region of the sky (i.e. the joint analysis of >400 images); optimizing the observing strategy and cadence of survey operations; identifying and mitigating systematic biases due to incomplete calibration of the data (e.g. for supernova cosmology); development of scalable algorithms that can measure the correlation function, and power spectrum on scales > 100 Mpc (i.e. to demonstrate the ability to detect Baryon Acoustic Oscillations); the development of optimized algorithms for the measurement of cluster masses; and strategies for characterizing supernovae light curves from noisy and incomplete data.

To understand and minimize systematic uncertainties within programs such as the LSST, high fidelity simulations of the data flow from these cosmological experiments have been developed. The results of this work are being used to design and test algorithms for use by the data management groups, to evaluate the capabilities and scalability of reduction and analysis pipelines, to test and optimize the scientific returns of the LSST survey, and to provide realistic simulated data from which to determine the scientific performance of the survey.

8.2.2 Computational Strategies (now and in 2017)

8.2.2.1 Approach

Image simulations: The LSST Phosim framework simulates astronomical images through a geometric ray-trace program that propagates photons through an atmosphere, telescope, and camera. Atmospheres are modeled using a Taylor frozen screen approximation with each screen described by a Kolmogorov spectrum. Photons are reflected and refracted by the optical surfaces within a telescope with mirrors and lenses simulated using geometric optics techniques in a fast ray-tracing algorithm. All optical surfaces include a spectrum of perturbations based on tolerances specified for a given telescope design. Fast techniques for finding intercepts on aspheric surfaces and altering the trajectory of a photon by reflection or wavelength-dependent refraction have been implemented to optimize efficiency. Ray tracing of the photons continues into the silicon of the detector with conversion probability and refraction (a function of wavelength and temperature) and charge diffusion within the silicon modeled for all photons. Photons are pixelated and the readout process simulated including blooming, charge saturation, charge transfer inefficiency, gain and offsets, hot pixels and columns, and QE variations. In this way, effects that might influence the optical performance of the telescope, as a function of wavelength and angle, can be rapidly evaluated.

Image Processing: Image processing pipelines need to perform near real-time calibration and analysis of acquired images. This includes transient detection and alert generation, annual processing of an entire data set for precision calibration, object detection and characterization, and support of user data access and analysis. For the LSST, images will be acquired at roughly a 17-second cadence, with alerts generated within one minute. Algorithm development will, therefore, address the dual requirements of efficient use of computational resources, and the accurate and reliable processing of the combination of deep and broad data resulting from the survey. This requires substantial progress beyond the state of the art from existing surveys. We anticipate the need for novel machine-learning algorithms for data quality analysis and to enable the discovery of the unexpected.

8.2.2.2 Codes and Algorithms

Image generation: The primary image simulation code, Phosim, is a fast ray-trace program that simulates the flow of photons through the atmosphere, telescope and into the camera. Computationally intensive routines are written in C++ with the overall framework and database interaction with the input catalogs using Python. The purpose of this design is to enable the generation of a wide range of data products for use by the collaboration; from all-sky catalogs used in simulations of the LSST calibration pipeline, to studies of the impact of survey cadence on recovering variability, to simulated images of a single LSST focal plane. The simulation framework is embarrassingly parallel with the unit of granularity a single CCD (each LSST focal plane comprises 189 CCDs and, during observations, one focal plane image is taken every 17s). No message passing is required between individual processors and the current version of code has scaled to over 50,000 cores without noticeable degradation in efficiencies.

Image Processing: The data management system for the LSST is written in Python and C++ (with computationally intensive codes in C++). Primarily developed for CPUs, subsets of the algorithms (principally image warping routines) have been ported to GPUs. The current framework includes the calibration and warping of images, detection of sources on images and the characterization of their photometric properties, the coaddition of multiple images, the subtraction of images, and the ingestion of detected sources into a database. The code is primarily a custom development that makes use of a number of open source libraries (FFTW, numpy, Boost, Mpich2, MySQL databases, Eigen, cfitsio, astrometry.net, Minuit2). The system has been tested running on 1,000 cores and will scale to 10,000 cores within the next year.

8.2.3 HPC Resources Used Today (2012)

8.2.3.1 Computational Hours

Image Simulations: For image simulation and analysis an initial request for NERSC compute time has been made for 2013 (with an award of 2M CPU hours). Earlier simulation runs (2M CPU hrs) have been undertaken on the Open Science Grid, the SLAC BaBar cluster, a Purdue Condor cluster, and using Google's Exacycle resources. These simulations have demonstrated that a single LSST visit (two back-to-back simulated focal planes) requires approximately 1,000 CPU hours to generate.

Image Processing: In 2013 we expect to use the 2M CPU hours on NERSC's Carver machine to generate 2000 LSST focal planes that will be used to evaluate current algorithms for measuring shapes of galaxies using repeated observations of the same part of the sky (up to 200 repeated observations). Processing and analysis of these images will be undertaken using a 1 million CPU hour allocation on the NSF's XSEDE resources.

8.2.3.2 Compute Cores

Image Simulations: The image simulation framework has been demonstrated running wide (many cores for a short amount of time) and deep (fewer cores but for longer simulation runs). On DOE BaBar resources (SLAC) Phosim has run on up to 2,000 cores. Using the Google Exacycle resources PhoSim has run on up to 60,000 cores. A limitation on the number of cores primarily comes from the initial IO required to load approximate 5GB of data at startup. Over the next 5 years it is expected that this startup IO will be reduced to < 1GB of data.

Image Processing: The data management framework has run on up to 1,000 cores with the primary bottleneck the ingestion of the data into a database on completion of the data run. The current configurations run at 1TB memory with 250 nodes, and 4GB per core.

8.2.3.3 Shared Data

Image Simulations: For 2013 we expect to generate 30TB of data that will be shared throughout the DESC and LSST collaborations. Data sharing to date has been accomplished using pull technologies from individual users including reddnet, rsync, and fast data transfer (FDT).

8.2.3.4 Archival Data Storage

Archival data storage has not been used due to the slow access patterns

8.2.4 HPC Requirements in 2017

8.2.4.1 Computational Hours Needed

Image generation: We expect to operate in two modes by 2017 (a) generation of images for rapid algorithm development whereby we simulate a representative set of focal plane images (typically 200 focal planes) with varying input data and observing conditions (b) data challenge data sets that correspond to the simulations of about 10% of a year's worth of LSST observations (~60,000 focal planes). We anticipate that the rapid generation process will occur monthly and the data challenge runs bi-annually. At 1,000 CPU hours per focal plane the required CPU hours would be ~120 million. To achieve this requires a doubling of the compute allocation for each year through 2017.

Data processing: The data challenge runs will dominate the processing and analysis requirements. At the current scaling of the LSST analysis algorithms it will require ~20 million CPU hours to process and analyze 60,000 LSST focal planes. This represents 10% of the final LSST data rate.

8.2.4.2 Number of Compute Cores

Image generation: Image generation runs in a pleasantly parallel mode (currently with no message passing). The image code has scaled up to 60,000 cores. For data challenges, the simulated data will need to be generated over a period <30 days. This would require sustained access to ~80,000 cores. Small-scale runs will require ~20,000 cores (in order to generate the data in <24 hours). It is likely that up to 3 concurrent small-scale simulation runs will be run by different DESC working groups.

Image Processing: Processing of the data is expected to run at ~20,000 cores and increase up to 60,000 cores (to evaluate the scaling of the processing system).

8.2.4.3 Data and I/O

Image generation: Each large simulation run will generate about 400 TB of data. Initial startup of the simulated tasks is expected to require ingestion of ~1-2GB of data per core.

Image Processing: Processing of the simulated data sets will read in 400TB of data and storing of the outputs will require ~800TB of data. For experimental data, ZTF will require a data rate of 15 GB/s for the analysis and classification of the data stream.

8.2.4.4 Shared Data

We expect to be able to serve the outputs from each of the large data challenges (i.e. 400 – 800 TB of data) to the DESC and LSST communities. The anticipated access patterns will involve large-scale transfer of complete data sets as well as the selection and delivery of small subsets of data. In a similar manner the outputs of the Nyx simulations (including mock catalogs) will require access from external users with the ability to extract subsets of the data.

8.2.4.5 Archival Data Storage

Each of the experiments will generate between 40 TB and 250 TB of data. All told, by 2017, we will have well over 700 TB of data in-hand. To allow for processing of this data as well as catalogs, etc., we estimate that we will require 1 PB of total storage.

8.2.4.6 Memory Required

Image generation: The Phosim code requires approximately 1GB per core.

Image Processing: Image processing is expected to require 1-4GB per core.

8.2.4.7 Software Applications and Tools

We need python, a C++ compiler, FFTW, numpy, Boost, MPI, OpenMP, ScaLAPCK, PSQL & MySQL databases, Eigen, cfitsio, astrometry.net, and Minuit2.

8.2.4.8 HPC Services

We will require Science Gateway Services (a la DeepSky) in order to share a variety of our results within the individual collaborations as well as to the outside community.

8.2.4.9 Time to Solution and Throughput

We expect to be able to keep up with the data processing in near real-time. To-date, the queues are such that this has not been a problem, but we will monitor the systems to make sure this does not change in the future and will work with NERSC to enable solutions to this unique HPC problem.

8.2.4.10 Data Intensive Needs

The cosmological simulations run under a classic HPC model. The image simulations and analysis applications are data intensive. Most tasks have a significant IO to CPU ratio (i.e. Amdahl numbers ~0.5). Application of these codes will likely range from embarrassingly parallel to small (1,000 core) MPI jobs. Most tasks will need to be run for tens of thousands of data sets or repeated multiple times using different initial conditions. For example, to generate simulated images for one night of LSST data requires the simulation of over 360,000 separate CCD images but each image can be simulated in isolation.

8.2.5 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	2 M	82 M
Typical number of cores* used for production runs	2,000	40,000
Maximum number of cores* that can be used for production runs	60,000	>60,000
Data read and written per run	6TB	250TB
Maximum I/O bandwidth	5 GB/sec	15 GB/sec
Shared filesystem space	20TB	250-500TB
Archival data	40TB	1PB
Memory per node	1GB/core	2-4GB/core
Aggregate memory	2TB	160TB

* "Conventional cores."

8.3 Cosmic Microwave Background Data Analysis

Principal Investigator: Julian Borrill
NERSC Repositories: planck, usplanck, mp107

8.3.1 Project Description

8.3.1.1 Overview and Context

Cosmic Microwave Background (CMB) experiments gather very large data sets whose analysis requires us to disentangle the various components (the CMB itself, astrophysical foregrounds, and instrumental noise) and their correlations, each of which is most simply expressed in a different domain (angular, spatial, and temporal respectively). In practice the analysis proceeds in 4 steps: making maps of the time-ordered data, separating the CMB and foreground components in the maps, estimating the temperature (T) and polarization (E and B) auto- and cross-spectra from the CMB maps, and estimating cosmological parameters from the power spectra.

The most computationally challenging elements of this process involve manipulations of the time-ordered data – both mapping real data and generating and mapping simulated data. The simulations are needed both to validate and verify our analysis tools and to quantify uncertainties and correct biases in our analysis of the real data. Ultimately we require up to O(10,000) Monte Carlo realizations of our data to constrain our uncertainties at the one percent level.

8.3.1.2 Scientific Objectives for 2017

The Planck satellite mission will continue through 2015, and the analysis of its data will continue beyond that. The primary goals will be to provide the definitive measurement of the CMB temperature anisotropies and the best possible characterization of the all-sky polarization – both of the CMB and of the foreground contaminants, and then to derive the tightest possible constraints on the fundamental parameters of cosmology from these. These results are assumed by all Dark Energy experiments to break degeneracies in their parameter spaces.

Beyond Planck, the next generation of suborbital CMB experiments (including EBEX and PolarBear) will focus on first constraining and ultimately detecting the B-mode polarization signal. At small angular scales this comes from the lensing of the E-mode polarization by intervening galaxy clusters, but at larger scales we expect to see a primordial signal imprinted by gravity waves generated during the inflationary epoch. Such a signal would be the “smoking gun” of inflation, would uniquely constrain its energy scale, and would likely result in a 3rd Nobel Prize for CMB studies.

The challenge for detecting the large-scale B-mode polarization comes from the faintness of the signal, orders of magnitude below the temperature fluctuations. As a result we require enormous data volumes to achieve the necessary signal-to-noise (up to 1000x the Planck data volume over the next 15 years), and exquisite control of systematic effects such as foreground contamination, instrumental effects (primarily beam asymmetry and band-pass mismatch), and analysis effects (specifically controlling leakage of T- and E-mode signals into the B-mode).

8.3.2 Computational Strategies (now and in 2017)

8.3.2.1 Approach

Our analysis is dominated by the need to find the maximum of a Gaussian likelihood function, but in a situation where the correlation matrix involved is dense with O(100,000,000) elements on a side. While we obviously cannot handle such a matrix explicitly, its inverse can be constructed as a product of sparse matrices so we can use approximate approaches where we only need to act with this inverse on a vector. However, such methods require Monte Carlo (MC) simulation sets to quantify their uncertainties and correct their biases. For publication-quality results we need to be able to simulate and map O(10,000) realizations of our data in O(1,000) wall-clock hours, while for intermediate tests we need O(100) realizations in O(10) hours; our general target then is O(10) realizations of the entire mission per hour.

To meet this requirement requires our codes to scale to very high concurrencies. To date, the bottlenecks to this have been their IO and communication costs; we have therefore focused on addressing these. We have successfully reduced these overheads to a point where we can run Planck-scale MC sets on up to O(10,000) cores and our largest individual map-makings on up to O(100,000) cores. We are aware of some remaining limitations to scaling the MC runs, however addressing these will require us to abandon the community code we are currently obliged to use for a dedicated in-house code designed to scale from the outset, and this work is currently in progress.

Beyond Planck, the suborbital datasets will provide a different challenge, constrained by calculation rather than communication or IO. For this we will need to develop our new code to take advantage of the next 10+ epochs of Moore's Law and 5+ generations of NERSC systems. In the first instance we will be pursuing the twin paths of extremely massive scaling (targeting many-core systems, including Intel MIC) and exploiting accelerators (primarily GPUs). We will also investigate better preconditioners for our conjugate gradient solver and sparse matrix-vector multipliers for our inverse noise weighting.

8.3.2.2 Codes and Algorithms

Our core code performs on-the-fly simulations (requiring parallel random number generation and Fourier/spherical harmonic transforms) and map-making (requiring a preconditioned conjugate gradient solver, with each iteration using sparse matrix-vector multiplication, Fourier transforms and a distributed map-reduction over all tasks). The current implementation is fully hybrid MPI/OpenMP.

8.3.3 HPC Resources Used Today

8.3.3.1 Computational Hours

Across all of the CMB projects we will use approximately 13 M MPP-hours at NERSC in 2012, with negligible allocations elsewhere. Had the resources been available on Hopper we could have used significantly more than this.

8.3.3.2 Compute Cores

Typical production runs currently use up to 30,000 cores due to known IO and communication bottlenecks in the community destriping map-making code that we are required to run for Planck. Our old maximum likelihood map-making code has been run on up to 150,000 cores, and its replacement – a hybrid destriping and maximum likelihood map-maker – is designed to scale to full systems.

Production runs currently incorporate O(10) instances of the code running simultaneously within a single job, and it is I/O contention reading data among the processes that prevents further multiplexing, and will be addressed by the next-generation code.

8.3.3.3 Data and I/O

Real data map-making runs need to read in the entire dataset at one frequency, which is on the order of 2TB. Currently it is much faster to first copy the data from the NGF shared file system to Hopper's scratch file system, a non-optimal process.
Full-scale Monte Carlo simulation/map-making runs need to write O(10,000) x 500MB maps, for a total of 5TB. These writes are also done to Hopper scratch and then the data are sync-ed to NGF.

8.3.3.4 Shared Data

Each major experiment supported by these repositories has its own NGF directory (planck, ebex, polar, quiet, spt); in addition we maintain a general NGF directory (cmb) for our software/modules. We find these shared data space invaluable for sharing data among members of the various teams. In sum, we have about 100 TB shared in project directories at NERSC.

8.3.3.5 Archival Data Storage

Roughly 550TB of data (primarily Planck and SPT, but increasingly PolarBear) have been archived in 2012.

8.3.3.6 Memory Required

Our peak total memory requirement is currently O(500) GB.

8.3.4 HPC Requirements in 2017

8.3.4.1 Computational Hours Needed

With multiple experiments each generating an order of magnitude more data than Planck – which itself has been under-resourced in 2012 - we can expect our computational requirements to increase 30-fold by 2017.

8.3.4.2 Number of Compute Cores

We expect our next-generation simulation/map-making code to break the current I/O and communication bottlenecks to scaling and be able to run on full systems in 2017.

8.3.4.3 Data and I/O

Real data map-making runs will need to read in the entire dataset at one frequency, of the order of 50TB, ideally without needing to mirror the data from NGF to a faster filesystem first. Full-scale Monte Carlo simulation/map-making runs will still need to write O(10,000) x 500MB maps, for a total of 5TB, again ideally writing directly to NGF.

8.3.4.4 Shared Data

With multiple experiments needing to store and share their data on NGF we will need on the order of 5PB of NGF space.

8.3.4.5 Archival Data Storage

At a minimum we will need 10x the spinning disk space on the archive, so O(50) PB.

8.3.4.6 Memory Required

Our code is designed to run on a very small per-node memory footprint if necessary, but 1-2 GB/core and would still be desirable. The aggregate memory required per run will be around 5TB.

8.3.4.7 Many-Core and/or GPU Architectures

Although we are not yet ready for GPUs, this is in our immediate development plan over the next 2 years, including using the Titan system at ORNL.

8.3.4.8 Software Applications and Tools

C, C++, Fortran (GNU) compilers.
Compatible MPI
Vendor math libraries (eg. libsci, acml, mkl)
Git

Everything else we build ourselves.

8.3.4.9 HPC Services

Account support
Consulting
Web interfaces
Federated authentication services
Dedicated hardware
Pseudo-user project accounts

8.3.4.10 Time to Solution and Throughput

Our major need is the ability to have fast job turnaround during working meetings and under deadlines for which the current boost process (high priority for jobs upon request) at NERSC works well.

8.3.4.11 Data Intensive Needs

It would be very useful to have tools that could be given a list of the data required by a queued-but-dependent job, pull it from hpss if necessary, and release the job dependency when that process was complete.

8.3.4.12 Additional Comments

Above all I need cycles, and sufficient infrastructure to get my data to them.

The most important thing NERSC can do is to provide reliable, stable, well-documented systems.

8.3.5 Requirements Summary Worksheet

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours	13M	500M
Typical number of cores* used for production runs	30,000	Full system
Maximum number of cores* that can be used for production runs	30,000	Full system
Data read and written per run	2TB read, 5TB write	50TB read, 5TB write
Maximum I/O bandwidth	10 GB/sec (from Hopper scratch)	250 GB/sec (from NGF)
Percent of runtime for I/O	35%	35%
Shared filesystem space	200 TB	5,000 TB
Archival data	550 TB	50,000 TB
Memory per node	1.2 GB	1-2 GB
Aggregate memory	0.5 TB	5 TB

- "Conventional cores."

8.4 Type Ia Supernovae

Principal Investigator: Stan Woosley
Worksheet Author (if not PI): Chris Malone
NERSC Repositories: m1400

8.4.1 Project Description

8.4.1.1 Overview and Context

Our research focuses on understanding the explosion mechanism driving Type Ia supernovae (SNIa), which were used to show that the Universe is accelerating in its expansion – research that won the 2011 Nobel Prize in Physics. Most of our attention on SNIa has been on the so-called single degenerate, Chandrasekhar-mass model where a massive white dwarf star accretes material from its companion star thereby compressing and heating the white dwarf's interior until thermonuclear fusion of carbon occurs in the core. The energy release from the carbon drives convection in the core for about a century until the extreme temperature sensitivity of the carbon-burning reactions causes a local runaway of burning and a flame is ignited. As the flame buoyantly rises towards the surface, the white dwarf expands and the overall density decreases. If the remainder of the star were burned as a flame at these lower densities the abundances of various isotopes in the ashes of the flame would not match observations; in particular, too many intermediate-mass elements are produced and not enough iron-group elements. Therefore, for all but perhaps the faintest of SNIa, the burning must transition at some point to a supersonic burning front: a detonation must occur. The exact nature of when and how this transition from a deflagration (flame) to a detonation occurs in an unconfined medium like a star is an open topic of research.

There are a few other models of SNIa in the literature on which our group has begun focusing. One candidate is the sub-Chandrasekhar-mass model where a white dwarf a little less massive than the Sun accretes a very thin layer of helium from a companion star. In a very similar fashion to the model described above, the helium at the surface becomes compressed and heated until a thermonuclear runaway forms. The conditions in the helium are much more likely to produce a prompt detonation, rather than a flame. This detonation of helium may then drive a shock into the underlying carbon/oxygen mixture of the white dwarf; this shock may then touch of a carbon detonation that incinerates the remainder of the star. Another model we have begun investigating recently involves the merger of two white dwarfs, each a bit less massive than the sun. This “double degenerate” model has several flavors that depend on the mass ratio between the two white dwarfs, magnetic field strength and the type of collision – head-on or off-center. In some cases a detonation is triggered very soon after merger, in others the explosion can be delayed.

The large range of spatial and time scales involved in all of the above scenarios demand high-performance computing. The thickness of a carbon flame in a white dwarf interior is on the order of 10^{-2} cm, whereas the radius of the star is on the order of 10^{8} cm. The convective phase leading up to ignition in the Chandrasekhar-mass model is about 10^{10} s, whereas the dynamical time of the explosion itself is on the order of a second. A simulation that resolves all relevant spatial scales and covers even a small fraction of the dynamic time range is simply impossible even on the largest computers of, perhaps, even the next decade. To combat this we usually use a spatial grid with coarse resolution (relative to the flame/turbulence scale) and approximately account for the physics occurring on the subgrid scale. We also employ one-dimensional models for large portions of the temporal evolution; we use the results of the 1-d models as initial conditions for our multidimensional studies of some small fraction of the evolution – for example, the last hours of the convective phase of the Chandrasekhar-mas model. As we attempt to more accurately portray the physics by increasing resolution or including more isotopes in our nuclear reaction network, the storage demands increase significantly. A *single* checkpoint file that stores the state information used to restart a calculation for our largest simulation to date is on the order of 300 GB. Storing and parsing many such files cannot be done on today's desktops or even most local clusters. Simulations on these scales require high-end computing.

8.4.2 Scientific Objectives for 2017

Over the next five years we plan to expand our studies of all three SNIa mechanisms mentioned above. For example, we will be adding more detailed nuclear reaction networks/tables using the latest compilation of reaction rates from the JINA ReacLib database. Using very large networks in the multidimensional simulations is not efficient as most of the time to advance the timestep is spent solving the stiff system of ODEs that govern the reactions. We will likely expand on our support of tracer particles that carry the local thermodynamic state information and can later be post-processed with very large reaction networks to get a more accurate nucleosynthetic yield. Another avenue currently being developed is the use of tables – also generated with large networks, off-line – to approximate the energy release and nucleosynthesis during a multidimensional simulation.

The simulations with more accurate nuclear physics will yield more accurate light curves and spectra, which, at the end of the day, are compared to observations of SNIa to rule-out or confirm the simulated explosion mechanism. There are a large number of supernovae observations, so we need a large number of simulations and light curves/spectra for a good statistical comparison. We plan to do many such simulations to create a library of models available to the public.

We will also be pushing to higher resolution simulations of the various aspects of SNIa evolution/explosion. This allows us to resolve some of the physical mechanisms that are currently on our subgrid scale model, and hence lets us relax some of our assumptions. In particular, we will likely investigate the turbulence-flame interaction in greater detail. This is very important in the Chandrasekhar-mass model where the turbulence from prior convection in the interior or from shear flows on the surface can alter the flame structure, boosting its burning speed, and possibly causing a transition to detonation.

8.4.3 Computational Strategies (now and in 2017)

8.4.3.1 Approach

Our codes use a finite volume approach, where the domain is decomposed into zones and we store average fluid quantities in each zone. We use Adaptive Mesh Refinement (AMR) to place higher resolution in regions of dynamical interest. For each zone we solve a set of conservation laws that relate the changes within a zone to the fluxes across the zone boundary and any sources or sinks within the zone.

Groups of zones – of size, say, 64^3 zones – at a single level are assigned to a single MPI task. Fine-grained parallelization is used by assigning several OpenMP threads to each MPI task. We are beginning to think about how to gain more performance by off-loading some portions of our codes to accelerators.

8.4.3.2 Codes and Algorithms

Maestro is an AMR hydrodynamics code designed for low Mach number astrophysical flows. The algorithm in Maestro approximates slow flows in hydrostatic equilibrium, filtering out the sound waves and allowing for stable timesteps that are larger than those allowed in traditional compressible codes. A constraint on the divergence of the velocity field captures compressibility effects due to background stratification and local heating/diffusion sources. Maestro uses a second-order approximate projection method to ensure the divergence constraint is satisfied; this is an elliptic solve computed using a multigrid method. Advection is handled using an unsplit Godunov method, and reactions are incorporated via operator splitting.

Castro is an AMR compressible hydrodynamics code designed for astrophysical flows. Castro and Maestro share the same underlying data structure (BoxLib) and parallelization scheme. Castro solves the compressible Euler equations along with self-gravity and nuclear reactions. Advection is handled using an unsplit Godunov method with either a piecewise linear or piecewise parabolic reconstruction. Newtonian self-gravity is incorporated as either a monopole approximation or as a solution to a Poisson equation using multigrid techniques. Nuclear reaction source terms are incorporated using operator splitting.

Sedona is a multidimensional, time-dependent, multiwavelength radiation transport code that calculates light curves and spectra of supernovae using an implicit Monte Carlo approach. Sedona also uses the BoxLib data format, and parallelization is done with both MPI and OpenMP. A significant portion of the compute time for Sedona involves calculating the wavelength-dependent opacities of supernovae debris. This involves reading in large tables of atomic data and calculating the populations of numerous atomic levels.

8.4.4 HPC Resources Used Today

8.4.4.1 Computational Hours

Our project used almost 13 million core hours in 2012 at NERSC. In addition, we used ~25 million core hours on an Early Science System allocation at NCSA's Blue Waters machine, and another ~28 million core hours at ORNL's jaguar machine before it was brought offline for upgrades to Titan. We've recently been awarded another ~20 million core hours at Blue Waters under their Friendly-User access program, which will be used through January 2013.

8.4.4.2 Compute Cores

We typically run production jobs at NERSC in the 128 – 24,576 core range depending on the type of scientific problem we are exploring. Our codes scale well out to ~100k cores, but production runs typically involve more complex systems solves that scale up to ~65k cores.

The number of cores used in a run is heavily dependent on whether the problem is 2d or 3d and the desired resolution for the particular physics problem we wish to study. If we are exploring a parameter space with small jobs, we typically have several running at a time.

8.4.4.3 Data

Our project has a NERSC project directory, which has about 3.5 TB of data currently. We also have about 100 TB stored on the NERSC archival storage system.

8.4.5 HPC Requirements in 2017

8.4.5.1 Computational Hours Needed

We expect that we will need 200 million hours at NERSC in 2017. This number assumes that we will receive additional large allocations at other facilities, such as ORNL, NCSA, and XSEDE.

8.4.5.2 Number of Compute Cores

A typical run in 2017 will likely use 50,000 cores and we will use up to 200,000. Our workflow will require having multiple concurrent runs.

8.4.5.3 Data and I/O

Per run our codes will typically write >~5 TB, read < 1 GB, and require a bandwidth greater than 10 GB/sec to disk. We want to keep our I/O time to 10 percent or less of the total run time.

8.4.5.4 Data

Our need for shared space in the NERSC project file system will increase to 200 TB and we'll need to archive 1 petabyte of data.

8.4.5.5 Memory Required

To date, we've not really had issues being memory-bound in our simulations; in addition, we are mainly restricted by memory per core, not per node. That may change, however, if we begin adding larger networks and higher resolution as outlined above. This also depends somewhat on the fine-grained parallelization strategy, and will likely be altered with the use of accelerators. That being said, the most memory-restrictive large-scale machine we have run on (hopper) had 1.3 GB of memory per core, and our code ran just fine. The current "standard" of 2GB/core is likely sufficient to handle our needs over the next five years; 1GB/core will probably be pushing the limits. Our largest science simulation to date (on Blue Waters) ran on 2048 nodes (65k "cores"; Interlagos architecture) and likely had < 60 TB in memory.

8.4.5.6 Many-Core and/or GPU Architectures

Currently we do not support accelerators in our codes. We plan to investigate this in the near future, mostly likely adding pragmas to our codes to offload some of the work to accelerators. It would be nice if NERSC keeps up-to-date with the current compiler suites that support these types of pragmas (OpenACC, next generation OpenMP, etc.)

Additionally, we expect that over the next 5 years or so the accelerator technology will be moving closer and closer to the chip. Hopefully, then, many of the programming difficulties inherent in highly optimizing our codes for attached (i.e. through a PCI bus) accelerators can be bypassed into a single framework for spawning tasks/threads amongst the various components. Ideally this framework will be transparent to the programmer. In the meantime, as stated, we will continue to investigate methods of leveraging the current state-of-the-art accelerator technology to realize our scientific goals.

8.4.5.7 Software Applications and Tools

We need the following software packages or technologies:

MPI, OpenMP, VODE, PETSC, FFTW, C/C++, Fortran95 or later, python, perl

8.4.5.8 HPC Services

Consulting/account support is always useful when something breaks or a user has login issues. For our very large data sets, we may see increasing interaction with the analysis/visualization team. Tutorials on handling large data sets will be useful.

8.4.5.9 Time to Solution and Throughput

We do not have any special needs.

8.4.5.10 Data Intensive Needs

We are going to be running into the "big data" problem fairly soon. Analyzing and visualizing data sets that are a significant fraction of a petabyte is going to be a challenge. Most of our data will stay local to the NERSC network, so transfer isn't so important, unless we are using something like grid-FTP to go to another computing center.

One area of need that has not been addressed is the possibility of doing in-situ or in-transit data analysis. Currently, our codes do not support this approach. Some assistance on how one goes about leveraging such technology would be beneficial. An ideal example would be if there were some sort of Map/Reduce-like mechanism for the large amounts of data we have in memory before a data dump that could help facilitate shrinking our data set to a more manageable subset for visualization/science extraction. This would likely involve both hardware and software initiatives.

8.4.5.11 Additional Comments

One concern with future systems is the lack of storage provided. Machines are getting ever faster, but the storage capacity is not keeping pace. For example, the upgrade to Titan at OLCF increased the FLOPS of the machine by about an order of magnitude, but the total aggregate scratch storage system has remained about the same size. This is probably mostly driven by technology limitations in industry for disk drives, but it is something to keep in mind for future systems.

8.4.6 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	13M	200M
Typical number of cores* used for production runs	128 – 24,576	50,000
Maximum number of cores* that can be used for production runs	65,000	200,000
Data read and written per run	< 1 TB	>5 TB
Maximum I/O bandwidth	~3 GB/sec	>~10 GB/sec
Percent of runtime for I/O	~15	~10
Shared filesystem space	~3 TB	200 TB
Archival data	~100 TB	1,024 TB
Memory per node	~20 GB	~64 GB
Aggregate memory	~30 TB	~200 TB

* “Conventional cores.” For GPUs and accelerators, please fill out section 4.7.

9 Energy Frontier Case Studies

The following case studies are representative of major research efforts in the Energy Frontier over the next five years.

9.1 Lattice Gauge Theory Calculations

Principal Investigators: Rich Brower, Steven Gottlieb and Doug Toussaint
Case Study Authors: Rich Brower, Steven Gottlieb and Doug Toussaint

NERSC Repositories: mp13, m1647

9.1.1 Project Description

QCD is the component of the standard model of sub-atomic physics that describes the strong interactions. It is a strong coupling theory, and many of its most important predictions can only be obtained through large-scale numerical simulations within the framework of lattice gauge theory.

These simulations are needed to obtain a quantitative understanding of the physical phenomena controlled by the strong interactions, to calculate the masses and decay properties of strong interacting particles or hadrons, to determine a number of the basic parameters of the standard model, to make precise tests of the standard model, and to search for physical phenomena that require physical ideas which go beyond the standard model for their understanding.

Lattice field theory calculations are essential for interpretation of many experiments done in high-energy and nuclear physics, both in the U.S. and abroad. Among the important experiments that have recently completed or are in the final stages of their data analysis are BaBar at SLAC, CLEO-c at Cornell, CDF and D0 at Fermilab, and Belle at KEK, Japan. New data is beginning to arrive from the LHCb experiment at the LHC, and BESIII in Beijing. In many cases, lattice QCD calculations of the effects of strong interactions on weak interaction processes (weak matrix elements) are needed to realize the full return on the large investments made in the experiments. In all such cases, it is the uncertainties in the lattice QCD calculations that limit the precision of the standard model tests. Our objective is to improve the precision of the calculations so that they are no longer the limiting factor.

Because the lattice approach to studying QCD in a nonperturbative way is so computationally expensive, the groups that create the gauge configurations have often made them publicly available. These configurations can be used for a wide variety of physics topics and by sharing their configurations with other collaborations the scientific impact can be maximized. The NERSC Gauge Connection was a pioneering service in support of sharing configurations worldwide and remains an important service of NERSC that relies on its storage facilities. Thus, both high-end computing and storage are essential to our research.

9.1.1.1 Scientific Objectives for 2017

QCD simulations proceed in two steps. In the first, one performs Monte Carlo calculations to generate ensembles of gauge field configurations in proportion to their weight in the Feynman path integral that defines the theory. These configurations are saved, and in the second step they are used to calculate a variety of physical quantities.

QCD is defined in the four-dimensional space-time continuum, but in order to perform numerical simulations, one must reformulate it on a lattice or grid. In order to obtain physical results it is necessary to perform calculations at a variety of lattice spacings, and perform extrapolations to the continuum (zero lattice spacing) limit.

The MILC collaboration is using the highly improved staggered quark (HISQ) action with four dynamical quarks to create ensembles of gauge configurations that will allow us to take the continuum limit with high precision. For 2017, we would like to create an ensemble with a lattice spacing of 0.03 fm, smaller than ever explored before. This small lattice spacing will be useful for calculation of a wide variety of physical quantities, especially for the study of b-quark decays.

We would also like to include the effects of fully dynamical quantum electrodynamics (QED) in these calculations. In current work, we are using quenched QED. The configurations that we generate will be used in a wide variety of physics studies. These include the weak decays of hadrons that can be used to study the CKM matrix that describes some of the fundamental parameters of the Standard Model (SM) of Elementary Particle Physics.

We will also study the masses of bound states of quarks including both mesons (quark anti-quark bound states) and baryons (three quark bound states). These will allow us to determine the masses of five of Nature's six quarks. These masses are also fundamental parameters of the SM.

9.1.2 Computational Strategies

9.1.2.1 Approach

Lattice QCD is a theory of quarks and gluons (gauge fields) defined on a four-dimensional space-time grid. We use a Monte Carlo method to create gauge configurations in proportion to their weight in the Feynman path integral that defines the theory. Once a suitable ensemble of configurations is created and stored, it can be used to study a wide variety of physical phenomena. The generation of configurations is a long stochastic simulation that must run at high speed. However, with the stored configurations, subsequent work can be done in parallel and the speed of a single job is not critical as long as there is sufficient capacity to run multiple jobs.

9.1.2.2 Codes and Algorithms

The MILC collaboration has developed its code over a period of 20 years and makes it freely available to others. It has evolved to match our physics goals and to accommodate changes in computers. Currently containing approximately 180,000 lines, it is used by several collaborations worldwide. Our code has made increasing use of a library of specialized data-parallel linear algebra and I/O routines developed over the past several years with support from the DOE's SciDAC program. These packages were developed for the benefit of the entire USQCD Collaboration, which consists of nearly all of the physicists in the United States who are working on the numerical study of lattice gauge theory. We are the principal developers of all but one of the C components of the SciDAC libraries, and their design was inspired by the MILC code. The MILC code has been used for benchmarking by SPEC, NERSC and for the NSF competition that lead to the Blue Waters project.

There are several algorithms used in the generation of gauge fields. The heart of the algorithm is a dynamical evolution similar to molecular dynamics. In order to calculate the forces driving the evolution, a multimass conjugate gradient solver is required to deal with the quarks. This solver takes the majority of the time in the calculation and it becomes increasingly important as the up and down quark masses are reduced. It has only recently become feasible to use up and down quark masses as light as in nature. In prior years, it was necessary to do calculations with a few heavier values for the mass and perform what is known as a chiral extrapolation.

The hypothetical case study that we use to estimate computational requirements for 2017 is a scaling up of a project that we are now running at NERSC and other centers. Thus, estimating time in Hopper-equivalent core-hours is fairly straightforward. In making the time estimates we have not assumed any algorithmic breakthroughs, although it is possible that advances such as application of multigrid techniques for sparse matrix solution will bring significant improvement. Although the time estimate is made with a particular choice of discretization of the underlying differential equations and a particular algorithm for the Monte Carlo sampling, we expect that the physical parameters of this sample problem will be what are needed in the 2017 time frame.

For this case study we describe a QCD simulation with lattice spacing of 0.3 femtometers (fm), with a physical spatial size of 7.5 fm. The simulation would contain four kinds of dynamical quark (up, down, strange and charm) with their masses tuned to their real-world masses. The simulation of light quarks with their real-world masses has only recently become feasible, but several groups around the world are now doing such simulations, and they will soon become expected for state of the art projects. Our time estimates assume that the Highly Improved Staggered Quark (HISQ) action is used to discretize the quark fields, and that the hybrid molecular dynamics method is used to evolve the system from one sample configuration, or lattice, to the next. The staggered quark actions are generically the least expensive to simulate, but it is possible that we or another group might choose to use a Wilson type quark discretization because of its simpler particle content, which would cost roughly a factor of four in the time estimates.

We are currently beginning a simulation with the quark masses at their physical values, using a lattice spacing of 0.06 fm and a spatial size of 5.5 fm. The proposed simulation uses a lattice spacing a factor of two finer. This will reduce most of the errors from the discretization by a factor of four. Also, it will greatly improve accuracy for properties of particles containing bottom quarks. With a mass of 4.5 billion electron volts, these quarks require careful treatment of the short distance, or high energy, physics. Currently the discretization errors from the bottom quarks on the lattice are a leading source of uncertainty in lattice computations of some of the fundamental parameters in the standard model. Using the same techniques we are now using, these heavy quark discretization errors would be reduced by a factor of two. Moreover, we expect that with a lattice spacing of 0.03 fm we will be able to treat the bottom quarks in the same way we now treat the charm and lighter quarks, with a likely large reduction of systematic errors. (It is worth noting that one group has already begun a simulation with a lattice spacing of 0.03 fm, although with a simpler fermion discretization, a much smaller spatial size, and unphysically heavy dynamical quarks, for precisely these reasons.) The somewhat larger physical size in our proposed project will reduce the systematic errors from enclosing the simulated system in an unphysical box, and as errors from discretization are reduced the other systematic errors must be reduced along with them, lest they come to dominate the overall errors. But we should also note that a spatial size of 7.5 fm will make the lattices more useful for other studies, such as nuclear physics, where two or more nucleons may require this much room to move around. Also, an emerging area of intense work is the studies of theories with different gauge groups and fermion types that the standard model, because these theories may hold the key to resolving some of the unnatural features of the simple theories of the Higgs particle(s). A generic feature of these theories is a "slow running" of the coupling constant, which means that the important behavior occurs over a large range of length scales. This means that a need for simulations in larger volumes can be expected in these projects just as in simulations of quantum chromodynamics.

9.1.3 HPC Resources Used Today

9.1.3.1 Computational Hours

The United States lattice gauge theory community, organized as the USQCD collaboration, operates computing clusters at DOE labs that currently provide around 300 million conventional core-hours per year, as well as several million GPU hours. In addition, about 100 million hours of time from the INCITE program at DOE computing centers is distributed among the US lattice gauge theory projects. In addition, several lattice gauge theory groups have allocations at local centers or at XSEDE centers. For example, the MILC collaboration has an allocation of about 42 M regular core hours and 0.8 M GPU hours at the XSEDE centers. In 2012 the collaboration used 75 million hours at NERSC.

9.1.3.2 Compute Cores

Our large production runs at NERSC today use either 24,576 or 18,432 Hopper cores. This size is chosen to give good turnaround in the NERSC queuing system and to take advantage of the discount for jobs over 16K cores. The code has been tested on up to 40K Cray cores with reasonable performance. The larger jobs that we envision for the future could make use of proportionally more cores, which, if the same number of lattice sites per core were used, would be 50 times greater.

9.1.3.3 Shared Data

We do have a project directory, although we don't use it for some of the things we probably should. It is called “milc” and currently stores just over 6 TB of data in 220,000 files.

9.1.3.4 Archival Data Storage

NERSC currently operates the “gauge connection” archive, which makes lattices produced in our earlier simulations available to the whole community. The MILC collaboration's HISQ action project currently does not use much archival space at NERSC. However, our future needs are quite uncertain because we have been using NSF resources for most of our archival storage, and it appears that NSF does not intend to keep providing multi-year storage. Thus our long term plans for archival storage are currently in a state of flux.

9.1.4 HPC Requirements in 2017

9.1.4.1 Computational Hours Needed

It is our hope that other continuation of the resources listed above will continue and expand, so that lattice gauge researchers will have a variety of ways to work, and an aggregate amount of computing power to continue our progress. To achieve our 2017 goals we will require the equivalent of about 160 billion core hours. Assuming that NERSC supplies 15% of that time – as it did in 2012 – we will need 24 billion hours at NERSC in 2017.

9.1.4.2 Number of Compute Cores

The volume of the lattice we have used in our hypothetical case study is fifty times the volume of the system we are currently running at NERSC, so it is reasonable to suppose that we can use fifty times as many cores. Whether these are conventional cores, something like the MICS architecture, or fewer cores with GPU acceleration, depends on the evolution of the various technologies.

9.1.4.3 Data and I/O

Each lattice in our case study problem is 100 GB, so a lattice generation program will need to read and write 100 GB, while an analysis program will need to read 100 GB, and may or may not write a similar amount of propagator data. Ideally, these lattices would be written in a time similar to what we are seeing now, or a few hundred seconds, requiring bandwidths approaching 1 GB/s.

9.1.4.4 Shared Data

A “lean” approach, where we frequently read and write to archival storage, would require 2-5 TB, while an easier-to-organize approach would require 10-20 TB.

9.1.4.5 Archival Data Storage

The hypothetical case study above will require around 200 TB of archival storage.

9.1.4.6 Memory Required

Note the 3 GB per node entered in the table is a rough estimate, multiplying a reasonable local lattice volume by an estimate of the number of vectors and matrices we have. It is strongly dependent on how many processes per core the eventual architecture will have.

9.1.4.7 Many-Core and/or GPU Architectures

We have been working since 2009 to prepare our codes for GPUs. This work will continue with NSF support. We started with the conjugate gradient solver for staggered quarks within the QUDA framework begun at Boston University. Currently, all of the major parts of the code needed for gauge field evolution with staggered quarks have been coded and tested on NVIDIA Fermi class GPUs. We are actively porting to Kepler and expect to have results there very soon. Our project on quenched electromagnetism has been carried out almost exclusively on GPU based systems including Forge and Greenstreet at NCSA, and Longhorn at TACC. We will be using Keeneland very soon. Gauge field generation has been benchmarked and we found that on the Dsg cluster at Fermilab, 128 GPUs out-performed 256 Hopper nodes by a factor of two using a 64-cubed-times-96 grid. There are further optimizations needed for our GPU code.

We are just getting started with the Intel MIC architecture and have access through the BEACON project. It is too soon to say how well this architecture will perform on our code and how extensively the code might need to be modified to run will on MIC. We plan to be studying this issue for at least six months.

NERSC could certainly help with this transition by providing training to our postdocs and graduate students for the new architectures. Of course, direct programming help would be even more valuable. For the GPU work, we worked closely with Guochun Shi, a computer scientist at NCSA. We had planned to hire him half time for the next three years to continue GPU development work, but he decided to work for Google. Two other key GPU developers have gone to work at NVIDIA. Thus, we are planning on relying physics students and postdocs for much of the future development. This is not always the best match in terms of personnel skill or career aspirations. Finding the right people is a serious concern right now. The same issue arises for MIC where we have one student assigned to that project.

9.1.4.8 Software Applications and Tools

We will continue to need standard parallel processing libraries, MPI, OpenMP and any successors. The USQCD community develops and maintains software for lattice gauge simulations, the "SciDAC libraries", and these will be needed. (We typically just install these in our own areas.)

9.1.5 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	75M	24,000M
Typical number of cores* used for production runs	24K	100K
Maximum number of cores* that can be used for production runs	100K +	1 M
Data read and written per run		1TB
Maximum I/O bandwidth		10 GB/s
Percent of runtime for I/O		10
Shared filesystem space	6.2 TB	20 TB
Archival data	23 TB	200 TB
Memory per node		3 GB
Aggregate memory		3 TB

* Traditional cores

9.2 Simulations Required for the Energy Frontier in High Energy Physics

Principal Investigator: Elizabeth Sexton-Kennedy, Robert Roser, Michele Papucci and Stefan Hoeche
NERSC Repositories: m1738

9.2.1 Project Description

9.2.1.1 Overview and Context

At a high level there are four classes of simulations required to support Energy Frontier research in High Energy Physics.

1. **Event Generators** calculate all of the Feynman diagrams for a particular physics process of interest and determine the 4-vectors of all the physics objects (leptons, baryons, mesons, bosons, etc.) for each collision. The results are written to a flat file.

2. **Detector Simulations** –typically GEANT4 – take output from the event generator as input and propagate particles through the different materials that comprise the detector. The detector model is crafted in great detail so the simulated detector response is identical to what the actual detector produces in real collisions.

3. **Event Reconstructions** take the hits from the real or simulated detector and determine particles trajectories, location, charge, momentum and energy.

4. **Physics analyses** loop over the reconstructed events, making choices to tease out the signal or measurement being sought after.

The LHC (Large Hadron Collider) is the primary tool of scientists working on the Energy Frontier. The most powerful atom smasher ever built, the LHC started operating in 2010 and completed its first data run at half energy in 2012. The program was designed to discover the Higgs (which it has done) as well as to search for new physics and "peel back the next layer of the onion" in our understanding of the quantum universe, therefore providing clues about open questions such as the nature of Dark Matter or why the gravitational force is much weaker than all the other known forces.

Currently there are two general-purpose detectors at the LHC – the Compact Muon Solenoid (CMS) and A Toroidal LHC Apparatus (ATLAS). These large and intricate devices are being used to make measurements that test fundamental parameters of the Standard Model of particle physics and search for physics beyond it. This is an extremely exciting time for physics at the Energy Frontier.

The recent observation of a resonance at 126 GeV this summer has raised as many questions as it has answered. If this resonance is the Higgs boson, why is it so light? Do we really live in a meta-stable universe where the fundamental parameters are fine-tuned?

Or is there some undiscovered symmetry that explains it? If such symmetry exists then there is a whole zoo of particles yet to be directly detected at the LHC. One of the possible scenarios, supersymmetry, may have its lightest new particle to be stable, heavy, neutral and not interacting electromagnetically. Such a particle, called the “neutralino”, would be a viable dark matter candidate, solving another mystery of our times. The Energy Frontier experiments are positioned to go after a deeper understanding of matter and the nature of space-time.

At the heart of the computing challenge of the LHC experiments are the small cross-sections of the interesting physical interactions as compared to the backgrounds. Discovery physics is always buried underneath a large standard model background that is not easily modeled. This means that a full spectrum of techniques from specialized triggering hardware, to massive amounts of commodity computing for storage and computation, are needed to sift through the 600 million collisions per second created at the LHC and to model the physics we do understand in order to separate out that which is new. In addition, the complexity of a single crossing, demands very large fine grained detectors leading to large recorded event sizes.

High Energy Theory research is mandatory in this effort, both for improving predictions for background processes and developing event generator software and for interpreting results presented by ATLAS and CMS for the wider panorama of new physics theoretical models. As the complexity of final states increases, providing precise theoretical predictions implies a significant increase in computation time. The needs are qualitatively similar to the experimental ones described above and are mostly focused on event generation and physics analysis.

9.2.1.2 Scientific Objectives for 2017

It is imperative that the LHC experiments investigate the properties and question raised by the discovery of the resonance at 126 GeV. Does it fully account for Electro Weak Symmetry Breaking? The answer relies on fully measuring its W, Z couplings and branching fractions to gauge bosons. The spin and parity have to be measured before we can confidently say this is a standard model Higgs. Does it couple to fermions? Are the fermion couplings proportional to the masses? Is this a singlet state or are there others higher mass states yet to be discovered? Are the measurements of all production modes as expected? Does it decay to new particles? Are there hints that it mixes with hidden sectors?

The next run of the LHC collider will be at a center of mass energy of 13TeV. Every time hadron colliders have run at higher energies there has been an important discovery. Given the implications of a light Higgs and the hints from cosmology, we know there must be physics beyond the Standard Model waiting to be discovered. Computing resources and code performance, will play a large role in the speed with which we can reconstruct and analyze these new and exciting results.

The LHC has entered a two-year shutdown period to prepare the magnets for high energy followed by another 2-year run. During the shutdown CMS and ATLAS will both do a final reprocessing of the data taken during the first collider run. They will continue to analyze and publish results, and prepare for the higher energy run by generating Monte Carlo samples. In addition we will have to develop and extensively test software to handle the challenging conditions we expect to be delivered in 2015.

HEP theory will develop new tools to increase the precision of calculated cross sections and simulated events. This involves QCD resummation at higher logarithmic accuracy, fully exclusive event generation at the next-to-next-to-leading order in perturbative QCD and the incorporation of electroweak effects into event generators. Theorists will also study in more detail the implications of the experimental results for the large set of models for new physics in the LHC energy range.

9.2.2 Computational Strategies (now and in 2017)

9.2.2.1 Approach

The primary usage model for LHC computing is best described as High Throughput Computing (HTC), not High Performance Computing (HPC), although high energy physics benefits from both technologies. What matters to us is the speed with which we can process the large datasets needed to do the science. The big advantage that HEP has used for decades is that each crossing is statistically independent of the next one. We have used this embarrassingly parallel nature of the problem to great advantage. We have placed large farms of Linux processors all over the world connected via high bandwidth networks between the sites. In a real way the nature of the problem has shaped the solution; what is not clear is how well this solution will scale into the future. CMS, with its current software configuration would require the memory per core ratio to remain as it is now, and memory bandwidth be maintained even for high core count nodes. However, development is going on to significantly parallelize our codes to gain speed. Note that the systems developed for HPC can be used for HTC. Fast interconnects are not needed for event generation and analysis, but they are advantageous for computing cross sections. Modern event generators perform the Monte-Carlo integral using adaptive sampling algorithms like VEGAS and the multi-channel method. The corresponding optimization procedure requires frequent exchange of information between different compute nodes, which benefits greatly from the high network bandwidth offered by HPC facilities.

9.2.3 Codes and Algorithms

As mentioned in the introduction, the LHC codes can be categorized into different problem domains.

1. The first is the simulation of scattering events. Stand-alone programs written either in FORTRAN or in C++ perform this. They may include Python API's and interfaces to external libraries like Root and LHAPDF. At this stage there is no input data. Calculations are performed as a Monte-Carlo integration using adaptive sampling algorithms. Programs for higher-order calculations make use of quadruple or arbitrary precision arithmetic. Some programs use POSIX threads,

OpenMP and MPI to parallelize computations. The simulation proceeds in two steps, the calculation of cross sections and the generation of events. Computing the cross sections can take between several CPU seconds for very simple processes and several CPU years for the most complicated ones. The output of this step is typically stored and reused in subsequent event generation runs, especially for high-multiplicity processes.

2. The next problem is to interface the event generators with the detector simulation. Theorists often tend to tailor their code to the local resources that they have access to, and this does not easily translate to the GRID environment. Some generators are written in C++, or are easily wrapped Fortran codes, which run fast enough to allow them to be run in the same job that simulates their interaction with the detector components of the experiment using Geant4, (about which there was a workshop earlier this year[2]). Other generators can run for many days producing very small text based output files. For these jobs the experiments use a separate workflow that reads in the text file, runs Geant4 and outputs a format that is similar to the faster generators. The time needed per event is a function of the type of hard scatter being simulated. In CMS it takes about 100 seconds/event for a typical top background event. In ATLAS, fully simulating an event takes about 3,300 HEPSpec06 (HS06) seconds, where a typical modern core has a rating of about 8-10 HS06 (i.e., translating to ~300 – 400 wall clock seconds). ATLAS also uses a less-detailed fast simulation where doing so doesn't compromise the physics; it takes about 310 HS06 per event.

3. The next problem is to digitize the response of the hit detectors including noise effects and the mixing in of hits from the many glancing collisions that happen in the same crossing as the hard scatter. The number of additional collisions is a function of the instantaneous luminosity. At 2012 luminosities this corresponds to adding 25-30 additional interactions, which makes this a data intensive problem. After digitization, the full reconstruction is run. The goal of the earlier steps in the simulation is to make the inputs to the reconstruction step look identical to the inputs from data collected from the detector, the "real data". In CMS it takes 20 sec/event to digitize and reconstruct the top background events. In ATLAS it takes about 830 HS06/event on average for digitization and reconstruction.

4. The fourth problem is to reconstruct the real data as promptly after the data is taken as possible and reprocess them as necessary. The algorithms that run in this process are combinatorial in nature, which makes the time taken by them a strong function of the number of hits recorded in the detector, which in turn is a function of the instantaneous luminosity. The different steps of reconstruction are mostly serialized. The pattern recognition in the tracking chambers must be done before the track fitting and particle flow algorithms can start. Calorimeter energy reconstruction and muon stub finding can run in parallel but those steps are relatively fast. 70% of the event reconstruction time goes into reconstructing

[2] See http://science.energy.gov/~/media/ascr/pdf/research/scidac/GEANT4-final.pdf

tracks. In CMS it takes 5 to 35 sec/event depending on luminosity. In 2012 the CMS average was 15sec/event. In ATLAS the average is about 230 HS06 per event.

5. The fifth problem is the end user analysis of the reconstructed and simulated data samples. Since some aspects of the high level reconstruction are analysis dependent, many analysts will redo some part of the event reconstruction during the filter selection and data reduction phase. There are a wide variety of data bandwidth and processor time requirements in this problem, however a typical CMS analysis job can take as much as 1sec/event. In ATLAS, analysis activity is divided into two broad categories: group analysis and individual analysis. In the former, physics working groups collaborate to run shared analysis production in the production system, producing a range of analysis data on output required by the working groups. Such analyses average about 20 HS06/event. In the latter case of individual analysis, individuals use the distributed analysis to (typically) produce small personal N-tuples, taking on average 0.4 HS06/event, but often running many iterative cycles.

6. Results reinterpretation done by theorists requires to run similar codes as experimental data analysis described above, applying multiple experimental analyses to a set of MC-generated events. Differently from the experimental analyses, the focus is on simulating larger sets of models and parameters, with an approximate model of detector effects. There is currently a large range of software types used for this task, with ongoing software development work to modularize and simplify the entire effort. Typical computation times vary between 1-3 event/sec, mostly limited by I/O speed in high-throughput mode.

9.2.4 HPC Resources Used Today (2012)

9.2.4.1 Computational Hours

We have not used any NERSC computing resources in 2012. (n.b., ATLAS researchers based at Berkeley Lab extensively use a Linux (named PDSF) housed at NERSC for detector simulation and data analysis. These researchers also use the NERSC data infrastructure.)

The LHC community uses resources of the Worldwide Large Hadron Collider GRID infrastructure (WLCG[3]), which includes regional GRIDs like the Open Science Grid (OSG) in the US and the European Grid Infrastructure (EGI) in Europe.

LHC jobs run continuously 24x7, they are not broken up into production runs. CMS and ATLAS typically run type 2 and 3 workflows as described above at their Tier-0, and Tier-1 sites due to the IO requirements of those jobs. Tier-2 sites run types 1 and 4. From the WLCG dashboard reports we can estimate the amount of computing used in the period between December 2011 and November 2012. The dashboard reports in terms of

[3] http://wlcg.web.cern.ch/

HEP-SPEC06 units (a benchmark specifically developed for HEP applications[4]), which can be converted to a cores count by picking a representative processor as a reference. Using the Intel Xeon E5520 as a basis, then each core delivers 10 HS06/s per core (if hyper-threading is deactivated, a little bit more than 10 HS06/core, if it is activated, a little bit less, so choosing 10 HSA06/core is a good estimate). Given this conversion the dashboard reports 40,800 and 16,300 Cores for ATLAS and CMS respectively at the Tier0+1s and 58,500 and 43,400 Cores for ATLAS and CMS respectively at the Tier2s used in parallel all the time. This sums to a total of 159,000 cores in continual use throughout the year, for a total of 1.39 billion core hours of usage.

Representative theory projects for the development, testing and validation of Monte-Carlo event generators use about 150k CPU hours per project with typically one or two such projects per month. Theoretical studies of LHC results require between 100k-400k CPU hours with typically at least 3-4 projects per year per PI.

9.2.4.2 Compute Cores

Each job typically uses a single core. If each core has two GBytes of physical memory, there is no limit to the number of cores our codes can use for production runs today. Work is going on to reduce this memory requirement under the expectation that future processors are likely to offer less memory per core, and in the hope that a reduced memory footprint will enable our codes to use a wider variety of computer architectures.

Modern MC event generators for theory use on the order of 128 compute cores per job and multiple jobs per production run. The maximum number of cores is limited only by network bandwidth. We have tested up to 1,536 cores in HPC mode so far.

9.2.4.3 Shared Data

CMS has a logical global namespace, which catalogs our files and datasets. There is a site-dependent physical namespace that maps onto the logical space, which serves as a translation table. At each site we store the files on disk or tape or both and the user specifies the logical file name, which gets transparently mapped for him/her when the job lands at a particular site. PhEDEx is the CMS transfer system, which manages replicas of files at the sites and transfers them among sites using grid FTP, SRM, and FTS. ATLAS data management is very similar, using the DQ2 distributed data management system (soon to be replaced by the next generation system Rucio). Right now we mostly send the jobs to the data, but in the future both experiments will move to jobs that access data across the wide area network.

9.2.4.4 Archival Data Storage

CMS and ATLAS have no data stored on the NERSC archive. (n.b., NERSC is an ATLAS tier-3 site, supporting research headquartered at Berkeley Lab.)

[4] http://hepix.caspur.it/benchmarks/doku.php?id=bench:results_sl5_x86_64_gcc_412

CMS will have 38 PB of tape storage at our Tier-1 sites by the end of 2012. CMS has a pledge of 47 PB and ATLAS has 38 PB at the Tier-1 sites.

Theory has comparably few storage requirements, with currently about 30 TB per collaboration for the development of MC event generators and 10 TB per collaboration for phenomenology.

9.2.5 HPC Requirements in 2017

9.2.5.1 Computational Hours Needed

We are currently do not have an allocation to use resources at NERSC. However, we are interested in exploring the possibility. We do expect to continue to receive computing resources from the WLCG.

When LHC data collection begins anew in 2015, we may need up to 10 times more computing power to keep pace with the collider's increased luminosity and data acquisition rate. However, we do not expect the size of existing (2012) ATLAS and CMS tier computing sites to change dramatically, although it will grow somewhat as older hardware is replaced as maintenance contracts expire.

Projecting the amount of compute hours we will need in 2017 is difficult given that it depends on several parameters that are unknown at this time. The most important of these is the instantaneous luminosity that the LHC will deliver in 2015, which could be twice the 2012 luminosity. Given the non-linear nature of the reconstruction time, we will not be able to continue with the reconstruction we are now using in workflows 2 and 3 above under these conditions. CMS and ATLAS also plan to take data at a rate that is 2.5 times the current nominal rate. If we are given the current luminosity conditions then we can predict the 2017 requirements by scaling with this factor since the uptime of the accelerator will likely not change. The number of events we collect is directly proportional to the amount of simulation needed and total compute hours required for both. If we are given the more challenging beam conditions of a 50-ns bunch spacing (with higher intensity bunches than the design 25-ns spacing) then with the same code we would need four times the computing due to the complexity of the events and 2.5 times due to the number of events for a total of 10 times more computing than in 2012.

Compute requirements for theory are expected to increase in 2017 due to the wide usage of higher-order perturbative QCD and the continuing development of resummation techniques. We would like to make extensive use of HPC facilities, if available, in order to reduce the turnaround time for tests and tuning of event generators. This means scaling up the current usage needs by roughly a factor of four to ten. Similar increase factors are also representative of theoretical analyses efforts, while the operation mode will likely remain embarrassingly parallel.

9.2.5.2 Number of Compute Cores

CMS is currently developing a multi-core framework that will be able to exploit both the conventional event level parallelism and a more fine-grained task based parallelism. We view this development as a hedge against a future where, either we will not be able to allocate 2 GB to each core, or memory bandwidth requirements will not scale, which will result in slowing down the processing. ATLAS similarly is presently validating in production a multi-core version of its framework, and has begun an intensive program of software optimization and rewrites over the next 2-3 years to be able to make maximal use of concurrency on new compute architectures.

MC event generators will likely use about 1k compute cores in parallel. The maximum will be limited by network bandwidth and by the fault tolerance of MPI. We will run multiple jobs concurrently and we expect the additional use of GPUs.

Again, there is no maximum number of conventional cores we can use. We can't use GPUs like those available today (see section below), although eventually we will be able to use a processor such as the Intel Phi, with small cores and big vector units. Investigating utilization of GPUs and other concurrency architectures is being pursued as an active R&D area in the experiment software communities.

9.2.5.3 Data and I/O

As noted above, our processes run completely independently of one another. The only I/O needed is for each job to read its input file (typically < 1MB/s for type-4 jobs or much less for type-2,3 jobs) and write its output to a local disk temporarily (similar rates as input for similar type jobs. After the job completes we would need to copy the data out to other sites.

9.2.5.4 Shared Data

We could store the software distribution in a shared space but we could also serve the code over the network using a caching server technology developed at CERN, the CERNVM File System (CVMFS). So in principle the answer is zero. However, each compute node would need to have temporary disk space to store output files before being transferred off site.

9.2.5.5 Archival Data Storage

We basically need all we can get. In fact, computing is moving to a model in which the archival data does not have to be co-located with the disk caches due to the availability of high speed wide area network connections. The LHC currently produces about 15 PB of data per year. By 2021 this rate could reach 130 PB per year.

9.2.5.6 Memory Required

Currently we need 2 GB per x86_64 core in order to run our applications. By 2017 we expect we will have properly multithreaded applications which may reduce this number, and should at least avoid its increase. We do not yet have good measurements on what will be required, but 1 GB per core would be a rough guess.

Theory expects shared memory requirements of 2-16 GB per compute core due to the increased complexity of calculations. It is already possible to reduce the size of required memory per node through multithreading, but threaded applications have intrinsic bottlenecks that are not easily removed due to the reliance on external libraries, like LHAPDF, which can only be operated in single-threaded mode at present.

9.2.5.7 Many-Core and/or GPU Architectures

Our software does not have a small number of clearly identifiable "kernels" as do other scientific applications; as such, it is less amenable to mapping onto GPUs. However, we do make many calculations that can exploit simpler SIMD-style vectorization, and to the extent that future processors provide such functionality (e.g., in Intel Phi) we should be able to exploit it at some level. As noted above, our current parallelization model is to run a single process per core. Other than the memory needs we don't see any specific issues using more lightweight cores. As part of moving to the multithreaded version of our framework, we expect to do the optimization work necessary to run on "lightweight" cores.

Several theory groups are working on making GPUs accessible for MC event generation, and some proof-of-concept implementations exist. We expect that by 2017 there will be some programs that make extensive use of GPUs.

9.2.5.8 Software Applications and Tools

In general our software requirements on compute nodes are fairly basic. We run our applications on standard Linux systems, usually RedHat Enterprise Linux (RHEL) and/or RHEL-rebuilds. We currently build on a RHEL5-rebuild (Scientific Linux 5), but typically have no problems running our applications on newer versions of RHEL, e.g., RHEL6. From the OS itself, we do not need any specialized kernel version and use only standard libraries shipped with RHEL, primarily standard glibc. Our applications are written in C++, with a bit of Fortran. We do use some specialized software or software versions (such as boost, a newer version of the compiler/libstc++, python, etc.) but we include all of this in our own software stack along with the software we write ourselves so as to keep the requirements we place on the run-time systems fairly simple.

In ATLAS we have the capability to deploy to sites using our own virtual machines if the site supports a virtual OS model. Based on the enthusiasm we see for cloud based computing among facility people, and our own positive experience with it, we expect to make increasing use of this deployment approach in the future. Should NERSC offer such access, ATLAS would make use of it.

Theory expects to use C++ and FORTRAN compilers, OpenMP, Python, Root and MPI, as well as Mathematica. Our build systems will need the GNU autotools.

9.2.5.9 HPC Services

The easiest way of interfacing NERSC systems to our applications would be to supply some front end or head node that looks like a WLCG system. It would require authentication services, batch system type scheduling, hooks for setting up the Squid caching server system we use to access calibration constants, and a mechanism for using our software distribution service.

9.2.5.10 Time to Solution and Throughput

As noted above, our computing problem is more properly classified as HTC than HPC, and consists of large numbers of independent jobs. Currently we run one process per core and the job runs to completion on typical time scale of *O*(8) hours. We have some flexibility to define shorter or longer jobs as necessary. By 2017 we expect that our applications will be multithreaded and be able to use many (x86_64) cores, perhaps up to the full number within a physical box. In CMS and ATLAS we are working to support short-lived resources like spot-market clouds and clusters that need to preempt quickly. The CMS goal is that a job can be evicted within 15 minutes, however it should be allowed to complete within a reasonable period of time since high throughput is the goal.

CMS typically has 75K to 100K jobs running concurrently on the GRID. ATLAS typically has 100K to 150K production and distributed analysis jobs concurrently running on the GRID.

9.2.5.11 Data Intensive Needs

We will need outbound connectivity for calibration constants or our own Squid servers within the HTC clusters. We would need to transfer out the output either directly or by storing and merging files on NERSC resources and then transferring them out. We might want to read input data through our federated network storage system (xrootd).

9.2.5.12 Requirements Summary Worksheet

CMS and ATLAS Computing and Storage Requirements Summary

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	continuous = 1,400 M hours	1,760-7,000
Typical number of cores* used for production runs	159,000 cores averaged over the year for ATLAS+CMS	200,000-800,000 cores
Maximum number of cores* that can be used for production runs	There is no limit	There is no limit
Data read and written per run (job)	.01TB	.02TB
Maximum I/O bandwidth	2GB/sec	10 GB/sec

Percent of runtime for I/O	Depends on type	
Shared filesystem space	45000TB+24000*2TB	233 PB
Archival data	76 PB	190 PB
Memory per core	2 GB	1 – 4 GB

Summary for Perturbative QCD and Phenomenology Computing
NERSC Repository: m1738

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours	0	15 M
Typical number of cores* used for production runs		10-30k
Maximum number of cores* that can be used for production runs		There is no limit
Data read and written per run (job)		2TB
Maximum I/O bandwidth		0.1 GB/sec
Percent of runtime for I/O		variable
Shared filesystem space	0	200TB
Archival data	0	500TB
Memory per core		2-16GB

*Conventional Cores

10 Intensity Frontier Computing

10.1 Introduction

[The following is excerpted from the 2013 particle physics Community Summer Study (aka "Snowmass") preliminary report[5].]

Computing at the Intensity Frontier (IF) has many significant challenges. The experiments, projects and theory all require demanding computing capabilities and technologies. Though not as data intensive as the LHC experiments, the IF experiments and IF computing have significant computing requirements in theory and modeling, beam line and experiment design, triggers and DAQ, online monitoring, event reconstruction and processing, and physics analysis. It is critical for the success of the field that IF computing is modern, capable, has adequate capacity and support, and is able to take advantage of the latest developments in computing hardware and software advances.

The IF has become the central focus of the US-based particle physics program. The transition to the IF dominated domestic program coincides with the transition at Fermilab from operating Energy Frontier (EF) experiments to operating IF experiments. Many of the IF experiments are designed to measure rare processes by using very intense beams of particles. Successful running of these experiments will involve not only the delivery of high intensity beams, but also the ability to efficiently store and analyze the data produced by the experiments.

The IF encompasses: 1) quark flavor physics, 2) charged lepton processes, 3) neutrinos, 4) baryon number violation, 5) new light weakly coupled particles, and 6) nucleons, nuclei and atoms. The requirements and resources of quark flavor physics, as in Belle II and LHCb, are more similar to those of the Energy Frontier. The requirements and resources of 4) and 5) are more similar to those of the Cosmic Frontier. We have thus maintained focused on the areas of charged lepton processes, neutrinos, baryon number violation and nucleons, nuclei and atoms.

Several experiments comprise the IF, including experiments to measure neutrino cross sections (MiniBooNE, MicroBooNE, MINERνA), experiments to measure neutrino oscillations over long (MINOS+, NOνA, LBNE) and short baselines (MiniBooNE, MicroBooNE), experiments to measure muon properties (g-2, μ2e), other precision experiments (SEAQUEST), as well as future experiments (ORKA, νSTORM). Each of those experiments represent collaborations between 50 and 400 people.

There is also strong US participation in several international IF experiments, such as Super-Kamiokande (SK), T2K, Daya Bay, SNO/SNO+ as well as US university lead experiments such as IceCube. The impact of the US contribution to the physics results of

[5] http://www.snowmass2013.org/tiki-index.php

these experiments is strongly correlated to the availability of computing resources and the efficiency of the computing model adopted. The groups participating in these experiments range in size from 30 to 250 people. In addition there is significant detector and experiment design R&D.

10.2 Resource Requirements

Typically the hardware demands of IF experiments are modest compared to those of the EF experiments. However, that does not mean that the needs are insignificant. For example, each experiment foresees the need of at least 1,000 dedicated grid slots per year for submitting jobs to batch processing facilities. In sum, IF experiments estimate needing 14,500 grid slots (about 127 million CPU hours) in 2017, compared to 7,650 used in 2013. Data storage needs are expected to grow from about 2.6 PB in 2013 to more than 8 PB in 2017.

Fermilab provides on-site computing resources, however they are not sufficient for all needs. University and other national lab resources are used for Monte Carlo generation. A common protocol to access these resources such as OSG is in the foreseeable future.

In addition to computing resources, there is a need for assistance with issues surrounding efficient data handling and script optimization. Resources for this are provided through Fermilab and would be extremely useful if increased.

International IF experiments in which U.S. physicists participate have significantly less support. U.S. groups have no dedicated U.S.-based grid computing resources. These experiments tend to rely either on resources in other countries, with low priority, or on university based resources that are shared amongst a broad pool of university users from multiple disciplines. As an example experiments like T2K run intensively on grid resources in Europe and Canada. Canadian and UK grid support was cited several times as a model both for grid computing and grid storage. These researchers must have access to dedicated resources that can be shared with other IF experiments in order to be competitive with analysis of data and simulation. It was widely noted that the lack of dedicated U.S. resources has a detrimental impact on the science.

The Daya Bay neutrino experiment, located in China used resources at NERSC, which was the Tier-1 center for the experiment. A summary of that experiment’s needs is given in the table below.

10.3 Requirements for the Daya Bay Experiment

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	1M	8M
Typical number of cores* used for production runs	400	3200
Maximum number of cores* that can be used for production runs	800	6400
Data read and written per run	140 TB	1,100 TB
Maximum I/O bandwidth	1 GB/sec	10 GB/sec
Percent of runtime for I/O	50	20
Shared filesystem space	500 TB	2,000 TB
Archival data	214 TB	4,000 TB
Memory per node	4 GB	8 GB
Aggregate memory	1,600 TB	128,000 TB

* "Conventional cores."

11 Accelerator Design and Simulation Case Studies

11.1 Community Petascale Project for Accelerator Science and Simulation (ComPASS)

Principal Investigator: Panagiotis Spentzouris
NERSC Repositories: m778 in 2012; m1646 beginning in 2013

11.1.1 Project Description

11.1.1.1 Overview and Context

Particle accelerators are critical to scientific discovery both nationally and worldwide. The development and optimization of accelerators are essential for advancing our understanding of the fundamental properties of matter, energy, space and time. Modeling of accelerator components and simulation of beam dynamics are necessary for understanding and optimizing the performance of existing accelerators, for optimizing the design and cost effectiveness of future accelerators, and for discovering and developing new acceleration techniques and technologies.

In the next decade, the high-energy physics community will explore the Intensity Frontier of particle physics by designing high intensity proton sources for neutrino physics and rare process searches. It will also be exploring the Energy Frontier of particle physics by operating the Large Hadron Collider, developing novel concepts and technologies necessary for the design of the next lepton collider (necessary for studying the properties of the newly discovered Higgs particle that will lead to new physics), and undertaking R&D for new acceleration technologies. The Community Project for Accelerator Science and Simulation (ComPASS) under SciDAC3 is developing the HPC tools and applications necessary to design Project-X, the proposed proton driver at Fermilab, and the next lepton collider, with either electron or muon beams, and with either conventional or advanced (plasma, dielectric structure) acceleration technology.

In the case of high-intensity accelerators it is imperative that beam-losses are kept under control. HPC resources are necessary to correctly model intensity dependent multi-particle and collective physics effects to identify and mitigate potential problems due to instabilities. The simulation runs incorporate multiple physics processes spanning a wide range of scales (multi-physics, multi-scale models) and are used to optimize and evaluate accelerator design parameters, resulting in an even stronger HPC resource requirement. In addition, integrated simulations (complete cryomodule, including higher-order modes and cryogenic losses) of the superconducting accelerating structures of the high-power LINACs required for such machines to study and control wake field effects involve very large numerical problems that can only be solved using HPC resources (see also the case study by K. Ko).

In the case of the next generation high-energy lepton colliders the key is minimizing size (thus maximizing acceleration gradient) and minimizing cost. The accelerator science community is pursuing R&D in different areas that show promise for achieving large acceleration gradients. ComPASS tools, under the SciDAC3 and INCITE programs, are already used to support this R&D in the following areas:

1. Plasma based acceleration, both beam and laser driven (motivation: plasmas are not subject to electrical breakdown that limits conventional structures). This technology will be important to scale beyond TeV energies for HEP and to provide brighter and smaller (laboratory- and hospital-scale) radiation sources. HPC simulations are essential due to the nonlinear, self-consistent evolution of the laser, beam and plasma response and are used to support experimental efforts at the BELLA and FACET facilities. Geddes and Tsung will also detail this topic in their respective case studies.

2. Utilization of lasers instead of microwave klystrons as the power sources because of the much larger intensities available from lasers. The challenge is to design a structure that can properly utilize the extraordinary power available from lasers to accelerate a charged particle beam. It is known that dielectric materials have higher damage thresholds than metals at optical frequencies, but require novel geometric structures for effective acceleration (photonic crystal waveguides, grating structures, etc.) Since Dielectric-Loaded Accelerating (DLA) structures are inherently more complex than conventional structures HPC is a necessity for the design process. Structures have feature sizes on the optical length scale but extend several wavelengths in each dimension, requiring tens to hundreds of millions of degrees of freedom to accurately resolve the physics. Fabricating these structures requires development of new manufacturing techniques, and many simulations are necessary to determine the effects of fabrication error.

3. Muon colliders are an attractive option because they are compact relative to e+e- colliders (no synchrotron radiation constraints); they could be cost-effective, with reasonable power consumption. In addition, development of a muon collider is synergistic with Intensity Frontier infrastructure, because they require intense proton beams to generate muons. This primary proton beam has to have short intense bunches at frequencies between 15 Hz and 60 Hz and power of ~4 MW. An upgrade to the planned Project-X facility will be necessary, and a bunch compression scheme designed. Given the challenge of the very high-intensities, required timing manipulation and other stringent beam requirements, HPC resources are necessary to model and help optimize such designs. In addition, collective effects need to be modeled for the cooling section of the accelerator complex, also demanding HPC capable codes and resources.

4. High-gradient using conventional structures: In order to achieve high-gradient acceleration at room temperature towards a multi-TeV e+e linear collider, a worldwide collaboration has been established for high-gradient R&D involving the development of new acceleration concepts and the design of various types of accelerator structures. The basic accelerator physics research includes the understanding of the gradient limit for structure operation without RF breakdown,

and the suppression of wake fields through appropriate damping mechanisms to maintain beam quality in a linear collider. The proposed Compact Linear Collider (CLIC) is based on a novel two-beam concept including a drive beam providing the acceleration power and a main beam to be accelerated. HPC is essential for modeling wake fields excited by the transit of an electron or positron bunch in the Power Extraction & Transfer Structure (PETS) of the drive beam LINAC and in the accelerator structures (AS) of the main beam LINAC (see also Case Study and talk by K. Ko).

11.1.1.2 Scientific Objectives for 2017

The 2017 ComPASS scientific objectives are driven by the major HEP community operation and R&D activities expected in 2017 in the Intensity and Energy Frontiers where advances in physics reach are enabled by advances in accelerator science.

Intensity Frontier Goals

The Fermilab Proton Improvement Plan (PIP) for neutrino and muon programs (to increase repetition rate and availability, minimize/control losses, and deliver ~3 times more protons per hour and ~2 protons per hour than current loss limited rates) is a key driver. PIP systems are scheduled for commissioning in 2016; ComPASS aims to provide multi-bunch (bunch to bunch physics effects), multi-physics (within bunch) models that are able to simulate the Fermilab Booster beam from injection to past transition (non-relativistic to relativistic particles). This is necessary for accurate loss prediction and instabilities, which is necessary for optimizing design and operation parameters.

The Project-X R&D and design effort will be continuing in the next five years. We plan to evaluate the higher order mode (HOM) wake fields of the superconducting RF (SRF) cavities and cryomodules to determine if HOM dampers are needed for the machine operation. The detailed analysis of the excitation of HOMs, both monopole and dipole, in the SRF cavities will be performed in the presence of cavity cell imperfection and misalignments and statistic variations to assess beam breakup conditions. In addition, we will study intensity-dependent effects that to lead to beam loss by exciting resonances formed by nonlinearities due to imperfections in the magnets and other structures in the accelerator lattice of the Main-Injector. To form realistic models of beam loss in the presence of these effects it is crucial to have a realistic model of the accelerator components, including the detailed apertures formed by them. We are also planning to incorporate detailed particle tracking to see the real effects of the physical apertures. The three classes of simulations we will perform are (1) sampling of random configurations of magnet imperfections; (2) operating-parameter scans to determine efficient machine working points under high intensities; and (3) trains of beam bunches coupled through wall impedance.

Energy Frontier Goals

ComPASS will continue to support the design effort for the next Energy Frontier machine, envisioned to be a lepton collider-based Higgs factory. The plasma driven acceleration R&D goals are described in detail in Geddes and Tsung's write-ups. In general, because of the goal to develop fully fleshed collider concept designs, new physics must be added to the models, as well as beam transport, scattering, and radiation effects. The high-gradient acceleration goals are detailed in Ko's writeup. We will extend our simulations to numerically quantify the dipole wake field cross coupling accurately in the realistic 3D geometry of the entire system of the two-beam module of CLIC (4 Power Extraction and Transfer and 8 Accelerating structures) to understand the intricate phenomenon and to devise measures to mitigate the effect. Due to the tight tolerances required in the machine design, simulation will also help us understand the effects of structure misalignments on the wake fields in the coupled structure.

For the muon collider option, the goal is to design the interface with the Project-X proton driver. The modeling requirements and advances necessary are similar those listed in the Intensity Frontier section for Project-X and PIP. In addition to these goals, the Muon Accelerator Program (MAP) has just begun the effort to utilize HPC resources in modeling collective effects in ionization cooling channels.

For the DLA option, the goal of the project over the next five years is to enable progress in all the challenging aspects of DLA development. This includes the experimental demonstration of acceleration over many Rayleigh lengths of a laser pulse (mm–cm) in several types of optical structures. Such experimental tests are taking place at the E163 facility at SLAC. Experimental research also includes improvement of fabrication techniques to more accurately match design geometries. The computational aspect of the research involves optimizing candidate structures for key accelerator parameters such as gradient and power efficiency, and understanding long-range beam dynamics.

11.1.2 Computational Strategies (now and in 2017)

11.1.2.1 Approach

ComPASS under the SciDAC program funds the development of a comprehensive toolkit for beam physics and plasma wave and electromagnetic structure design and optimization. Depending on the type of problem, different algorithms and approaches are employed: electrostatic (multigrid, AMR multigrid, spectral), electromagnetic (finite element direct and hybrid, extended stencil finite-difference, AMR finite-difference), quasi-static (spectral). Particle in Cell (PIC) techniques are employed in most cases, where depending on the physics of the problem, domain decomposition, particle decomposition, or hybrid decomposition is used. There may be communication of particle data, grid data, or both; some codes use a particle manager and some do not. In summary, PIC techniques, particle field solvers, linear algebra solvers and eigensolvers are the most commonly employed strategies in the ComPASS toolkit. The codes of the ComPASS toolkit are listed below.

11.1.2.2 Codes and Algorithms

ACE3P is a framework providing many capabilities for modeling of electromagnetic structures. It includes the OMEGA3P code (frequency domain) and T3P code (time domain). (1) OMEGA3P: 3D Maxwell Eigensolver for finding normal modes in lossless and lossy RF cavities using higher order finite elements on tetrahedral grid by solving linear or nonlinear large-scale eigenvalue problems. The numerical techniques include exact Shift-Invert Lancozs/Anordi, Second Order Arnoldi, Nonlinear Arnoldi, Nonlinear Jacobi-Davidson, and Self-Consistent Iteration methods with WSMP/MUMPS/SuperLU/iterative linear solvers. (2) T3P: 3D time-domain Maxwell solver using finite element discretization in space and implicit Newmark-beta scheme in time to solve the 2nd order vector field equation. It simulates the transient field response in RF structures due to excitations by imposed fields, dipole or transit beam.

OSIRIS is an object-oriented, fully explicit PIC code written in Fortran 90 and MPI. It advances the electromagnetic fields using the Maxwell's equations and advances the particle orbits in time using Newton's equation. One advantage of the object-oriented programming style is that it allows for modular and therefore safer and faster developments of new features. Over the years, a variety of features have been added to it, making it an amazingly versatile tool for a variety of problems in plasma physics. OSIRIS has been ported to many platforms and is being used in many groups around the world for a large variety of plasma problems.

QuickPIC is a highly efficient, fully parallelized, fully relativistic, three-dimensional particle-in-cell code for simulating plasma and laser wake field acceleration. It solves a reduced (quasi-static) description for the Maxwell's equations, where a fully three-dimensional electromagnetic field solve and particle push is reduced to a two-dimensional field solve and particle push. QuickPIC is constructed by embedding a parallel 2-D PIC code inside a parallel 3-D PIC code. The 2-D piece of the code mostly solves Poisson type equations with diffusion, which requires an iteration loop. Overall this algorithm speeds up the computational time by two to three orders of magnitude without losing accuracy for problems of interest. The underlying mathematical formulations are the Newton- Lorentz equation for charge particle's motion and the reduced Maxwell's equations in Lorentz gauge. The reduced Maxwell's equations are Poisson's equation for the scalar and vector potentials. The quasi-static model has instability that requires modification to the Poisson solver to include a diffusion process that is combined with an iterative solve. The reduced Maxwell's equations are solved in Fourier space with either periodic, conducting boundary condition using FFT (or FST, FCT). We use Boris pusher to update the particles' trajectory. The complete set of the equations are not time-centered, therefore they require a predictor-corrector scheme.

Synergia is a multi-language, extensible framework utilizing state-of-the-art numerical libraries, solvers, and physics models. Synergia features 3-D space-charge and impedance modules, and arbitrary order Lie maps for magnetic optics. Selected features include multi-bunch, ramping and RF and magnet, multi-turn injection, and active feedback modeling. Synergia uses a quasi−static model of the beam and calculates space−charge effects self−consistently. It employs multiple Poisson solvers including an FFT-based solver and a multigrid solver. It is used to model both proton machines (linacs and rings) and lepton machines. Synergia uses the split operator technique for tracking to separate magnetic optics and collective effects. For magnetic optics utilizes Lie algebraic techniques in conjunction with automatic differentiation. For the collective effects we utilize particle in cell techniques with FFT (uniform grid) and finite difference multigrid (non-uniform grid) based Poisson solvers.

VORPAL is a classical particle-in-cell (PIC) code that uses a structured Cartesian Yee mesh and a charge-conserving Buneman current deposition, with a 2nd-order Boris particle advance. Using sophisticated template metaprogramming techniques of C++, VORPAL uses the same source code for operation in one, two or three spatial dimensions. Field-induced tunneling ionization and electron-impact ionization algorithms are included. VORPAL can also model charged, relativistic fluids or warm neutral fluids and includes a DSMC (direct simulation Monte Carlo) algorithm for neutral particles.

Warp is a both a code and a general purpose framework for parallel 3-D PIC simulations of beams in accelerators, plasmas, laser-plasma systems, non- neutral plasma traps, sources, and other applications. It contains multiple field solvers (electrostatic FFT, multigrid, electromagnetic), internal conductors (cut-cell method with electrostatic solver), surface physics (space-charge limited emission, secondary emission of electrons or gas from impact of electrons or ions), volumetric ionization. It employs advanced methods such as cut-cell boundaries, Adaptive Mesh Refinement, and boosted-frame capability. Elaborate initialization and run-time options allow realistic modeling of experimental setups. Warp can couple to the electron cloud buildup code Posinst for providing fully self-consistent modeling of electron cloud effects, using a quasistatic solver for the coupling of particle beams and electron clouds. Warp parallelizes problems using domain decomposition in 1D, 2D or 3D, with the MPI protocol providing communication between processors.

11.1.3 HPC Resources Used Today

11.1.3.1 Computational Hours

ComPASS SciDAC development efforts used 3.8 M hours at NERSC in 2012. ComPASS codes are also used for applications in the repositories discussed by Geddes, Ko, and Tsung. ComPASS researchers also have ALCF allocations (5M hours, to become 80M hours in 2013). In addition, the Muon Accelerator Program (MAP) that pursues muon collider R&D is just starting utilization of HPC activities. They estimate ~13M hours at NERSC for 2013, and 25M hours for 2017 (this is also discussed in section 12.4 and included in the table). NOTE: in 2013, because of the new SciDAC3 project, the m778 repo became m1646.

11.1.3.2 Compute Cores

Maximum and typical processor usage varies by application type, code and dimensionality of run. Typical 3D PIC runs utilize 16k to 32k cores. The larger problems could utilize 64k cores. In general, the range is determined by problem size and queue constraints during a particular run. For the largest runs, parallel I/O begins to become a bottleneck, requiring special attention to run setup at > 10kcores. We often have of order ten runs executing concurrently.

11.1.3.3 Shared Data

We use the p-pwfa project directory, which currently has about 120 GB stored in it.

11.1.3.4 Archival Data Storage

We currently have about 30 TB stored.

11.1.4 HPC Requirements in 2017

11.1.4.1 Computational Hours Needed

Here the assumption is that the status quo remains, and the ComPASS repo is complimented by relevant INCITE awards for related applications (both at NERSC and ALCF). If we focus on Intensity Frontier applications the need for parameter optimization, and multi-bunch, multi-physics simulations increases the size of the problem by at least a factor of ten. Given the above, ComPASS will require 60M conventional compute hours. The MAP program estimates 25M hours. This will probably be in a separate repository, but the number is added at the table here, since they do not have a separate case study.

11.1.4.2 Number of Compute Cores

Here we focus on description of jobs that only run under ComPASS (see Geddes, Ko, and Tsung for other types of jobs). The multi-bunch FNAL Booster simulations require 84 bunches. We expect to average 600K steps per simulation. We can utilize roughly 512 cores per bunch, so we expect the runs to be jobs of roughly 40K cores (the bunch-to-bunch physics processes calculation is included in the core count per bunch). The FNAL Main Injector parameter scans will be loosely coupled groups of 64-128 jobs using 1024-2048 cores each. These jobs take roughly 2M steps. The multi-bunch Main Injector jobs can run for shorter simulated times, thus will only require 800K steps. These are the largest jobs in this category: 588 bunches at 128 cores/bunch will require over 75K cores per job, with the high-resolution jobs requiring 512 cores/bunch, for accurate beam-tail modeling, thus requiring 300k cores for a high resolution job.

11.1.4.3 Data and I/O

About 10 TB are written per run for Intensity Frontier applications. Data I/O is a major issue for scaling of many of the codes we run (see Geddes and Tsung). We presently observe scaling up to the few thousand processors, with a parallel I/O bandwidth of 0.3 GB/s. As the size of the job increases, writing files from individual processors is not an attractive alternative. A hybrid strategy may be required to solve this issue, with the necessary support, such as tools and queues to post-combine files. We would like to keep I/O below 20% of compute time.

11.1.4.4 Shared Data

We expect to have ~100 TB of shared data in 2017.

11.1.4.5 Archival Data Storage

At least the parameter optimization results for Intensity Frontier applications will have to be archived, to be used for comprehensive analysis: 500 TB

11.1.4.6 Memory Required

Without knowing the number of cores per node this is difficult to estimate. Our codes principally require a memory space of ~1 GB per MPI task or OpenMP thread running on each core. Hence the memory per node required scales with the number of cores per node. Some of the finite element codes in our toolkit might have larger requirements (see Ko's writeup).

11.1.4.7 Many-Core and/or GPU Architectures

The ComPASS research team has been working toward optimizing our code suite for upcoming architecture changes on future production HPC systems, either with "lightweight" processors, accelerators, or hybrid. Our strategy is to abstract and parameterize our data structures so that are portable and enable efficient flow of data to a large number of processing units in order to maintain performance. We believe that this strategy will enable us to deploy our codes quickly as soon as the target architecture is defined (it is apparent that next generation HPC resources will rely on SIMD processors where the main bottleneck is the memory bandwidth)

Although it is not clear whether the next supercomputer will be a many-core architecture or one based on GPUs, the next generation supercomputers will use SIMD processors where the main bottleneck is the memory bandwidth. Our strategy is to develop a portable data structure that will maintain the flow of data to the large number of processing units in order to maintain good performance. We believe this strategy will work for all of the next-generation processors. Even with such a "pro-active" approach, it will be very important to give sufficient notice and specifications of the chosen new NERSC architecture and provide a test system well in advance of commissioning the new machine. Such a test system will allow us to port and optimize our codes with the new compilers and relevant software environment.

We are already applying our development strategy on currently available "new" architectures. For example, we have developed a GPU-enhanced version of Synergia using CUDA. The GPU-enhanced portions include the 3-D open boundary condition space charge module and the basic set of bunch diagnostics. The current GPU version is limited to single beam bunches. The performance on a single processor with a single CPU currently exceeds the strong-scaling limit of a 128-node Linux cluster with Infiniband interconnects. Performance using a four-GPU system and a communication avoidance scheme is a little better than twice the single-GPU performance. During the next five years, if such systems are available for production, we could extend the

Synergia CUDA implementation to include all of the available space charge solvers and all of the diagnostics. In addition, we will extend the code to work on multiple bunches. With multiple bunches, Synergia has already been shown excellent weak-scaling behavior up to 1024 bunches, so we expect to be able to efficiently utilize a few thousand GPUs. Another example is VORPAL, which can currently perform EM computations on GPUs with metallic boundaries, and development of dielectric models for GPUs is expected in the next few years. Finally, the UCLA research team has developed a sophisticated PIC framework that could be used as a base for development on most new architectures, hybrid, "light"-core, or accelerator based.

11.1.4.8 Software Applications and Tools

Most libraries are built along with the codes. We use FFTW, HDF, LAPACK, METIS, MPI, MPI/IO, MUMPS, ScaLAPACK, SuperLU, TAU, TAO, Trillinos, shared libraries, and Python. High performance use of HDF5 and python are particularly important. For analysis we use ParaView, Python, R Language, ROOT, VisIt. The availability of long serial queues for analysis and data file processing/combination is vital. It would be very useful for run scheduling tools and monitoring to be enabled which would allow users to see and administer jobs without being subject to SSH auto-logout.

11.1.4.9 HPC Services

Support for performance optimization (tools and consultants) for the new system and in particular new architectures is important.

11.1.4.10 Time to Solution and Throughput

We need better queues to allow for development and production. Bigger jobs in a debug queue for a limited time are needed to debug many-core issues.

11.1.4.11 Data Intensive Needs

See the Data and I/O section above.

11.1.5 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	6 M	85 M*
Typical number of cores used for production runs	16 K	75 K
Maximum number of cores that can be	64K	300K

used for production runs		
Data read and written per run	1TB	10TB
Maximum I/O bandwidth	0.3 GB/sec	3 GB/sec
Percent of runtime for I/O	20	20
Shared filesystem space	10 TB	100 TB
Archival data	30 TB	500 TB
Memory per ~~node~~ core	1 GB	GB
Aggregate memory	1 TB	10 TB

* Includes MAP estimate for muon collider.

11.2 Laser Plasma Accelerator Simulation

Principal Investigator: Cameron G.R. Geddes
NERSC Repositories: m558

11.2.1 Project Description

11.2.1.1 Overview and Context

Laser-plasma acceleration (LPA) of charged particles shows great promise for reducing the cost and size of next-generation electron and positron accelerators for the DOE high energy physics program and accelerator stewardship. Plasmas are not subject to the electrical breakdown that limits conventional accelerators; accelerating gradients thousands of times those obtained in conventional accelerators have been obtained using the electric field of a plasma wave (wake field) driven by an intense laser. Such accelerators will be important to scale beyond TeV energies for high energy physics and to provide brighter and smaller (laboratory- and hospital-scale) radiation sources including free-electron lasers, Thomson sources, and ultrafast THz.

Simulations are essential due to the nonlinear, self-consistent evolution of the laser and plasma response. Key physics questions the simulations must address range from the dynamics of plasma ionization and formation to prepare the target, to laser propagation and energy transfer in meter-scale plasmas, and the injection and evolution of high quality particle beams in the plasma and laser fields.

The project supports a rapidly growing experimental and theory program, recently including demonstration of high quality beams at 1 GeV using a 3-cm plasma and stable beams with controllable energy. It also supports experiments now in progress on staging multiple LPAs on the BELLA PW laser, now operating and targeting 10 GeV in 1 meter. The separation of spatial and time scales between the micron laser period and the centimeter to meter scale acceleration distance means these simulations stretch current computational capabilities. The core methods are explicit and implicit particle in cell and fluid models as detailed below.

11.2.1.2 Scientific Objectives for 2017

The BELLA PW laser is now operational at LBNL to drive experiments on 10-GeV LPAs in meter-scale plasmas, and this together with experiments on staging of multiple LPAs to reach very high energies and injectors to create the required high quality bunches are the focus of the next five years. Related goals include light sources driven by LPAs, such as free-electron lasers and gamma ray sources, and the development of a user-facility level LPA test stand accelerator that requires advances in accelerator design for reliability.

Initial simulations of 10 GeV stages have been made possible by combining Hopper resources with envelope and Lorentz boosted PIC simulations, but resolution is allocation constrained. It has recently been demonstrated that LPAs can produce very low transverse emittance, and it is crucial to accurately model these and future beams, which will require three to ten times greater resolution in each axis. At the same time new physics must be added to the models as greater fidelity is demanded. We must accurately model plasma formation using MHD tools as well as beam transport, and must include scattering and radiation contributions to beam evolution. Positron production and capture, and beam conditioning, will be important to develop collider concepts as these accelerators advance. Addressing these needs is the computational focus over the next five years. This is required to accurately design and understand efficient high quality one to ten GeV LPAs, and to control staging of multiple modules and transport of low emittance bunches required for collider and other applications.

The LPA field is growing rapidly, and other large laser facilities are coming on line in the same time frame. Electron beam driven plasma accelerators are also being developed, including the now operational FACET facility at SLAC, as detailed in the case study by Tsung et al. The key accelerator physics and computational needs are similar across these projects.

11.2.2 Computational Strategies (now and in 2017)

11.2.2.1 Approach

The accelerator simulations model propagation of the laser through a plasma, plasma evolution and influence on the laser, the resulting formation of a plasma wave accelerating structure, and the injection and acceleration of a particle beam. We use explicit particle in cell (PIC) or fluid codes that resolve the laser oscillation period, and envelope codes that average over this period. Explicit finite difference time domain PIC codes (including Lorentz boosted simulations) parallelize via domain decomposition and scale well currently to more than 10k cores. An important limit is I/O, which typically does not scale well and must be addressed. Envelope codes scale to several thousand cores. Multidimensional Vlasov codes are another attractive approach, which so far has not been possible with current computers, but may become applicable soon.

Plasma ionization, formation and shaping are modeled using MHD fluid codes. Over the next five years these codes will be extended to 3D and parallelized and will require NERSC resources. Such codes typically scale to thousands of cores. Radiation of the particles is modeled with a separate code that implements particle tracking and interpolation of trajectories in order to resolve short-wavelength radiation. Because each trajectory is separate, this code scales very well to thousands of cores. The radiation is calculated from the trajectories and projected to a virtual detector. Over the next five years, many of these methods together with scattering will be integrated into plasma codes.

11.2.2.2 Codes and Algorithms

WARP is a code and a general-purpose framework for parallel three-dimensional PIC simulations of beams in accelerators, plasmas, laser-plasma systems, non-neutral plasma traps, sources, and other applications. For this project, it is used principally in explicit, electromagnetic PIC mode. It contains multiple field solvers (electrostatic FFT, multigrid, or electromagnetic), internal conductors (cut-cell method with electrostatic solver), surface physics (space-charge limited emission, secondary emission of electrons or gas from impact of electrons or ions), and volumetric ionization. It employs advanced methods such as cut-cell boundaries, Adaptive Mesh Refinement, a "warped" coordinate system with no paraxial assumption nor reference orbit required, and boosted-frame support, to name a few.

VORPAL is a parallel framework for finite-difference time domain (FDTD) simulations of charged species, represented as fluids and/or particles, in electric and magnetic fields, including a wide range of possible boundary conditions, algorithmic approximations, inter- species collisions and field-induced ionization. Particle-in-cell (PIC) techniques are used for charged particles, while fields and fluids are represented on a variety of structured meshes. VORPAL has been used successfully for several types of physical problems: laser- plasma interactions, RF cavities with complicated geometry, beam-structure interactions, RF plasma interactions, and basic plasma phenomena such as dynamical friction and anisotropic Debye shielding. For this project, VORPAL is used to model laser-plasma interactions and the associated electron acceleration. These simulations are typically electromagnetic (i.e., the full set of Maxwell's equations are used) with relativistic particles. The electron plasma is usually represented by particles via PIC, but can also be represented as a cold, charged fluid. For some problems, the laser fields can be represented approximately by an "envelope" model, with a corresponding ponderomotive particle push, for dramatically reduced run times. Field-induced tunneling ionization is also included for certain problems.

INF&RNO (INtegrated Fluid & paRticle simulatioN cOde) is a 2D cylindrical (r-z) code to model the interaction of short laser pulses with an underdense plasma. The code is based on an envelope model for the laser field, and the action of the laser on the plasma is modeled with the time-averaged ponderomotive force. Either a PIC or a (cold) fluid description can be used to model the background plasma. Both PIC and fluid modalities are integrated in the same framework, allowing for staged simulations. The code features an improved laser envelope solver that enables an accurate description of the laser pulse evolution deep into depletion even at a reasonably low resolution. A Lorentz-boosted-frame modeling capability for the fluid modality is also available, allowing for a significant speed-up in the calculations. The code is parallelized exploiting 1D and 2D domain decomposition. Compared with standard simulation tools, INF&RNO allows for a reduction of many orders of magnitude in computational time (between 2 and 5), while retaining physical fidelity.

VDSR is a parallel, object-oriented code for particle tracking and radiation calculation. Two kinds of calculation models are used for classical and quantum radiation. The trajectory of the particles in the beam can be traced either given the external fields or uploaded from the output of standard PIC codes (i.e. VORPAL or VLPL). A virtual detector records the radiation emitted by each particle and sums it incoherently with the radiation from other particles. Since every particle is independent the parallelization is performed in the particle domain and each processor calculates only a subset of particles.

As detailed in the case study by Tsung, et al., another repository for plasma accelerators has similar requirements and structure to those outlined here. Under that repository, OSIRIS is the explicit electromagnetic PIC code, QuickPIC is a envelope quasistatic code.

11.2.3 HPC Resources Used Today

11.2.3.1 Computational Hours

We used 12 Million hours at NERSC in 2012.

11.2.3.2 Compute Cores

Maximum and typical processor usage varies by code and dimensionality of run. Explicit PIC runs are typically 500-1,000 cores in 2D and 5,000 – 16,000 cores in 3D. The range is determined by problem size and queue constraints during a particular run. For the largest runs, parallel I/O begins to become a bottleneck, requiring special attention to run setup at more than 10k cores. Envelope runs use up to a few thousand cores. We often have of order ten runs executing concurrently.

11.2.3.3 Shared Data

Project directory incite7 is used on this project and currently uses 2 TB of space in 140,000 files.

11.2.3.4 Archival Data Storage

Currently, about 160 TB is stored on HPSS.

11.2.4 HPC Requirements in 2017

11.2.4.1 Computational Hours Needed

Increasing resolution to resolve low emittance beams and simulating tens of stages will require a minimum of 30x current resources. Simulation of a full 100 stages would require ~100x resources. This implies use of order 500-1,000 million hours.

11.2.4.2 Number of Compute Cores

Since many of our goals involve increased problem resolution/size, weak scaling dominates and cores will increase 10-30x, from 50k to potentially as much as 500k depending on the problem. This will also depend on interconnect and I/O performance to enable scaling. There will also be strong demand for tens of concurrent jobs using a few thousand-core problem sizes for parameter optimization.

11.2.4.3 Data and I/O

Data I/O is a principal issue for scaling of the codes we run. We presently use HDF5 parallel I/O and observe little or no scaling of I/O beyond the few thousand processor level, which means the I/O fraction of compute time increases linearly with number of processors. We see bandwidth of order 0.3 GB/s, which limits scaling. An alternative is to write individual processor files, but this also becomes difficult to manage at tens of thousands of processors. I/O is presently a few percent of time and the aim is to keep it at, or below, 10%.

Data I/O per run will expand to the 100-TB level. The amount of data that can be written will depend on I/O performance attainable and this is a key driver of result usefulness. Bandwidth above 10 GB/s would match needs. Three GB/s could be tolerated but would require more aggressive data subsetting.

If the required bandwidth can be achieved using parallel I/O such as HDF5 then that will be most efficient. If not, other projects (e.g. Cosmology, see Nugent & Borill) have developed custom I/O strategies (such as boxlib) that have sufficient performance (>30GB/s) for our 2017 needs, but these require specialized readers, etc. Common tools for these formats and queues to post-combine files, are needed. Inline analysis and data subsetting will also need to be improved and used.

11.2.4.4 Shared Data

We anticipate having 600 TB of shared data by 2017 unless I/O rates limit our ability to write simulation data.

11.2.4.5 Archival Data Storage

We will have to archive 5,000 TB unless I/O rates limit our ability to write simulation data.

11.2.4.6 Memory Required

Without knowing the number of cores per node that the hardware will contain this is difficult to estimate. Our codes principally require a memory space of ~ 0.1 GB per core or MPI task. In the future, hybrid OpenMP/MPI approaches may be used which may change the ratio of memory per core to memory per task. However, in the 5-year timescale, memory needed is anticipated to remain close to 0.1 GB/core. Hence the memory per node required scales with the number of cores per node.

11.2.4.7 Many-Core and/or GPU Architectures

Lightweight cores should function well with PIC codes, subject to memory needs as described above. We have experimented with our codes on GPU test beds at NERSC and on a GPU cluster at Northern Illinois University. In general the methods can be run on

GPUs and development is being pursued. AVX also appears to be attractive. However, at this point each system requires specialized development. Hence, high priorities to make a GPU system productive would include having sufficient notice of the chosen architecture with a test bed available (a year or more in advance) to allow porting of codes, and to have compilers that can handle, as much as possible, the loop ordering and related optimizations. PIC and related codes are not compute intensive, having particle push times that are typically microseconds/particle on CPUs, and GPUs can give more than 10x speedup. Hence maintaining reasonable bandwidth to and from the network for the GPU is a high priority.

11.2.4.8 Software Applications and Tools

Most libraries are built along with the codes. We use MPI I/O, HDF5 parallel, Trillinos, Mecurial, shared libraries, and python. High performance use of HDF5 and python are particularly important. We use VisIt and IDL for analysis. Having long serial queues for analysis and data file processing/combination is vital. It would be very useful to have run scheduling and monitoring tools that would allow users to see and administer jobs without being subjected to SSH auto-logout. This might include a system that is read-only or has limited privileges to address security concerns.

11.2.4.9 HPC Services

Support for optimization of codes on the machine, in particular for new architectures, has been and will continue to be needed. I/O support is an area of particular need. General consulting to assist users on the machines is an area of strength for NERSC and should also continue to be supported. Parallel visualization and analytics are an area of ongoing collaboration and will be increasingly important, since existing serial tools will not cope with the increased problem sizes anticipated. Support for run administration tools that may involve specialized authentication services, would be very beneficial, as detailed above.

11.2.4.10 Time to Solution and Throughput

Increased flexibility for premium queues would be a topic to explore: for example, more than one level of priority or cost. To prevent abuse, it may be worth allowing only a certain fraction of jobs to be premium, or accounting for the fraction of premium jobs as part of allocations.

11.2.4.11 Data Intensive Needs

As noted above, I/O is a primary scaling issue for these codes. We do not have other data intensive needs.

11.2.5 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	12M	500M[6] 1000M[7]
Typical number of cores used for production runs	5,000	50,000
Maximum number of cores that can be used for production runs	16,000	250-500k
Data read and written per run	3 TB	100 TB
Maximum I/O bandwidth	0.3 GB/sec	10 GB/sec
Percent of runtime for I/O	10	10
Shared filesystem space	2 TB	600 TB
Archival data	160 TB	5,000 TB
Memory per node is determined by number of cores per node. Memory requirement is per core.	0.1 GB per core	0.1 GB per core
Aggregate memory	0.5 TB	20 TB

11.3 Continuing Studies of Plasma Based Accelerators

Principal Investigator: W. B. Mori (UCLA)
Worksheet Author: F. S. Tsung (UCLA)
NERSC Repositories: mp113

11.3.1 Project Description

11.3.1.1 Overview and Context

A plasma wake field accelerator (PWFA) uses a particle beam to drive a plasma wave, which in turn accelerates electrons (or positrons) to very high energy. PWFA, like its Laser Plasma Acceleration (LPA) counterpart, uses the driver to excite highly nonlinear plasma waves, which in turn can be used to accelerate electrons. The UCLA simulation group is working with several experimental groups from around the world, using both electron beams (at the FACET facility in Stanford) and proton beams (at CERN and Fermilab) as drivers to accelerate particles to very high energies. The details of these experiments and the outstanding scientific issues involved are described below.

A. Studies on FACET Experiments and PWFA Collider Concepts

PWFA uses a particle drive beam to excite a wake field inside the plasma in order to accelerate another trailing beam. The plasma wake field can provide an accelerating field on the order of 10 GV/m as well as the focusing fields when the trailing beam stays on the right phase. Due to the energy conservation law, a high-energy gain (e.g., 10 GeV) on the trailing beam usually requires that the drive beam have an initial energy on the same order. FACET (Facilities for Accelerator Science and Experimental Test Beams) at SLAC aim to use a 23-GeV electron (or positron) drive beam to demonstrate PWFA ideas. One of those experiments is the two-bunches PWFA (Fig. 1), which will verify a high-energy gain (~ 10 GeV) on the trailing beam while maintaining a narrow energy spread. In this experiment an electron drive beam will excite a nonlinear plasma wake field in which the space charge forces of the drive beam can expel all the plasma electrons within a radius greater than or equal to the plasma skin depth, leaving a bubble-like cavity around the drive beam filled only with ions. A second electron beam is properly loaded inside this bubble just behind the drive beam. The accelerating field felt by the trailing beam can be flattened due to a beam loading effect, which will result in a narrow energy spread. The PIC codes QuickPIC and OSIRIS can self-consistently simulate such experiments and guide the experiments as well.

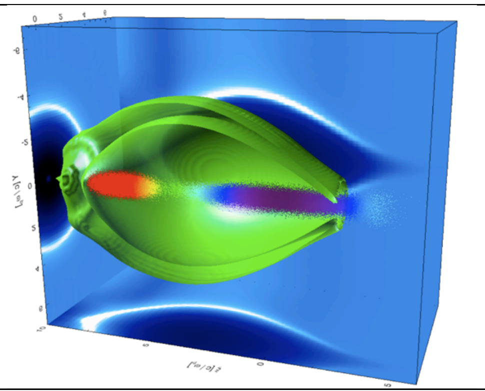


Fig 1.: Plot of a two-bunch PWFA in the "Blow-Out" regime from a QuickPIC simulation. The plots on the walls (in blue) are the cross-section of the plasma electron density along the x = 0, y = 0 and ξ = 0 planes. The green isosurfaces of the plasma electron density show the structure of the bubble sheath. The colored dots are two beam particles with different energy (blue represents low energy and red represents high energy). The two bunches are moving from left to the right.

The final goal for the research at FACET is to build up a linear collider using PWFA, with a much higher accelerating gradient (~1000 times greater) than the conventional accelerator. In the linear collider design, the beam has a very small emittance (~0.1 mm mrad in one transverse direction). As a result, its matched spot size inside the plasma wake field is ~ 100 nm. With these parameters, more issues should be considered. One of those is the ion motion issue. With a spot size around 100 nm, the beam density is high enough to cause significant ion motion during the beam passing through the plasma. The ion motion will break the linearity of the focusing force in the wake field and lead to the beam emittance growth. Simulations with such parameters become challenging due to the resolution requirements. QuickPIC has been improved to satisfy the needs. With the simulation study the plasma ion motion and beam emittance growth can be characterized. In addition, positron acceleration will also be investigated with the linear collider beam parameters in the future.

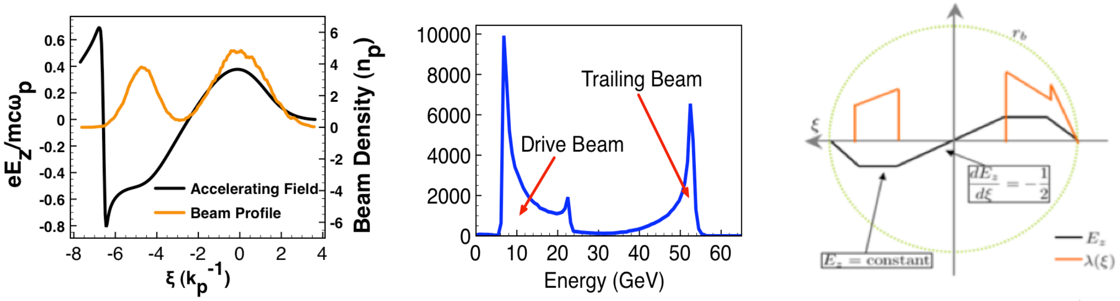


Figure 2: Left: Two bunch current profile and a lineout of the accelerating field for a possible FACET experiment. Middle: Predicted output energy spectra after 1 meter of propagation. Right. Current profile for an optimized design for a PWFA-LC.

A typical PWFA linear collider simulation (with 1 meter long propagation distance) will need **1 million** CPU hours using the code QuickPIC. A typical 3D QuickPIC uses 137 billion grids (using 2048x8192x8192 grids) and 30 million beam particles and 268 million plasma particles (per 2D slice).

B. Proton Driven PWFA

Proton-driven PWFA has gotten a lot of attention recently. As mentioned previously, PWFA for high energy purposes will need at least a 10 GeV drive beam. The highest energy particle beam that exists today is the 7 TeV proton and antiproton beams at CERN, which makes them attractive sources for driving a PWFA device. But these proton beams have much longer pulse lengths (~ 10 cm) compared to the plasma skin depth (~ 100 μm). Such beams may have self modulation (Fig. 2 and 3) when propagating inside the plasma, which can lead to the micro bunching of the beam and excite large amplitude plasma wake fields. In our studies, we will use a 120 GeV proton beam, which is an existing beam at Fermilab in the US. The first step is to simulate the proton beam self modulation with different beam parameters and plasma densities in order to find the self modulation condition and characterize the plasma wake field and the beam evolution. Osiris (in 2D cylindrical coordinates) and QuickPIC (in 3D) will be used for the self-consistent simulations on this problem. The next step is to investigate accelerating a witness electron beam using a proton driven plasma wake field. Other issues like hosing instability and filamentation instability will also be studied in the next few years. The studies will also support proton beam self-modulation experiments at Fermilab.

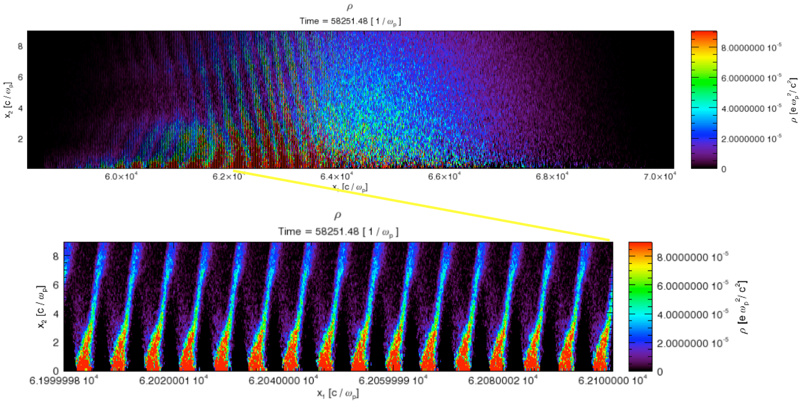


Fig. 3: Osiris 2D simulation of proton beam self modulation. The beam's total particle number is N = 1 x 10^{11}; the spot size is σ_r = 100 μm; the pulse length is σ_z = 10 cm; the emittance is ε_N = 3.33 mm mrad. The plasma Denstiy: n_p = 1 x 10^{16} cm^{-3}. The beam's focal plane is at 64 cm from the entrance of the plasma. The propagation distance is 2.13m.

A typical proton driven PWFA simulation will cost **150,000** CPU hours using the code QuickPIC. A typical 3D QuickPIC simulation uses 4.3 billion grids (262,144 grids in the longitudinal direction and 128x128 grids in the transverse direction) and uses 10 billion beam particles. A typical 2D OSIRIS simulation uses 120,000 cells (in z) by 90 cells (in r) and uses 20 particles per cell (216 million particles total).

11.3.1.2 Scientific Objectives for 2017

PWFA issues that will be investigated in the next few years include:

- The drive beam head erosion in the field ionized plasma.
- The hosing instability in the PWFA.
- The effect of an asymmetric drive beam.
- Dark current in current FACET experiments.
- Accelerating positrons using either an electron drive beam or a positron drive beam.
- Multiple Stage PWFA's
- Self modulation instability associated with proton driven PWFA's

11.3.2 Computational Strategies (now and in 2017)

11.3.2.1 Approach

The PWFA problem is solved mainly using two codes, OSIRIS and QuickPIC. OSIRIS is a fully explicit PIC code that can be run in 2D cylindrical (r,z) coordinates and 3D, while QuickPIC is a 3D PIC code that uses the quasi-static approximation for the beam (or laser) drivers. Both OSIRIS and QuickPIC have shown good scaling for > 10^4 cores. **OSIRIS was chosen as one of the codes for the 2011 Joule Metrics benchmark and it achieved > 30% peak speed (.74PFlops) on the full Jaguar machine.**

For 2017, we expect that the fully explicit and the quasi-static models will be sufficient to study the PWFA problem; however, we plan to port these codes to run on new manycore and GPU architectures. In Figure 4 we show the speedups of OSIRIS using SSE vectorization.

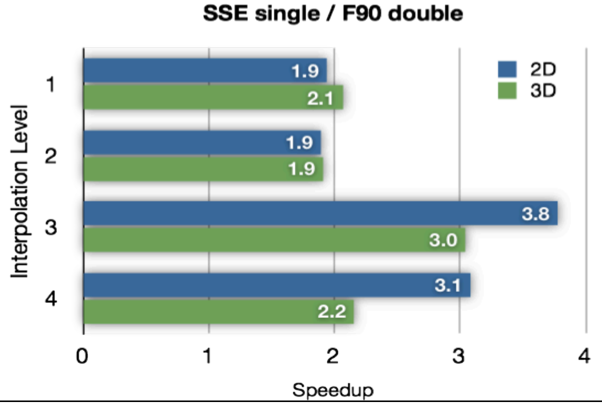


Figure 4: Speedup of OSIRIS using single-precision SSE vectors (blue bars indicate the speedup of the code in 2D while the green bars show the speedup of the SSE code in 3D. In all the cases, the SSE-enabled version of OSIRIS achieved a speedup of 2-3 over those that do not take advantage of the SSE extensions.

11.3.2.2 Codes and Algorithms

OSIRIS is the state-of-the-art, fully explicit, multidimensional, fully parallelized, fully relativistic, PIC code. Parallelization is done using domain decomposition with MPI. One of OSIRIS' strongest attributes is a sophisticated array of diagnostic and visualization capabilities that have also been ported into several of our other codes.

The OSIRIS framework includes 3D dynamic load balancing, an OpenMP/MPI hybrid decomposition, higher order particle splines together with current smoothing and compensation for improved energy conservation, a diagnostic which accumulates and tracks the trajectories for a pre-selected group of particles, and perfectly matched layers for transmitting light out of the simulation domain. In addition, new physics packages beyond those in standard PIC have been added. These include tunnel and impact ionization as well as a relativistically correct binary collision operator.

OSIRIS has been modified to take advantage of the SSE vector extensions. In our tests, the SSE version of OSIRIS achieves speedup of 2-3 over the non-optimized version. A GPU-enabled version of OSIRIS is also currently being developed.

QuickPIC: QuickPIC is a highly efficient, fully parallelized, fully relativistic, three-dimensional particle-in-cell model for modeling plasma and laser wake field acceleration. The model is based on the quasi-static or frozen field approximation, which reduces the electromagnetic field solve and particle push from 3-D to 2-D. This is done by calculating the plasma wake assuming that the drive beam and/or laser does not evolve during the time it takes for it to pass a plasma particle. The complete electromagnetic fields of the plasma wake and its associated index of refraction are then used to evolve the drive beam and/or laser using very large time steps. This algorithm reduces the computational time by two to three orders of magnitude compared to fully electromagnetic PIC codes. QuickPIC has shown good scaling to thousands of cores using the pipelining algorithm.

11.3.3 HPC Resources Used Today

11.3.3.1 Computational Hours

In 2012 the UCLA simulation group used over 27 million CPU hours on the Hopper supercomputer at NERSC and the Jaguar supercomputer at NCCS/ORNL (8 million on Hopper and 19 million on Jaguar/Titan). Of those 27 million hours, 15 of those were used for the study of plasma wake field accelerators.

11.3.3.2 Compute Cores

For the large 3D problem described in section 2, a typical run takes ~15,000 cores.

11.3.3.3 Shared Data

The project does not currently have a shared project directory at NERSC.

11.3.3.4 Archival Data Storage

As shown in the table, each simulation generates about 20TB of data, of which, 15 – 17 TB are for restarts (checkpoints) and 2-5 TB are for results. We expect to archive five simulations per year and therefore we estimate the annual HPSS usage to be ~10 TB. As of the of 2012, we had 90 TB of data stored in the NERSC HPSS system.

11.3.4 HPC Requirements in 2017

11.3.4.1 Computational Hours Needed

We expect the computational requirement to increase 20 fold in 2017. Therefore we expect to use ~300 M hours in CY 2017. Some of the physics that can be addressed by the increase in compute hours include:

- The drive beam head erosion in the field ionized plasma.
- The hosing instability in the PWFA.
- The effect of an asymmetric drive beam.
- Dark current in current FACET experiments.
- Accelerating positrons using either an electron drive beam or a positron drive beam.
- Multiple Stage PWFA's
- Self modulation instability associated with proton driven PWFA's

11.3.4.2 Number of Compute Cores

Due to the increase in the size of simulations, we expect to use ~300,000 conventional cores in 2017, using 15-20 TB of memory. In theory the OSIRIS code communicates only to the neighbor nodes and can scale to arbitrarily large number of cores; it has already shown good (>80%) strong scaling for ~300,000 cores on the Jugene supercomputer.

11.3.4.3 Data and I/O

We expect to generate 50 TB of data per run, including ~15 TB for checkpoint/restart. A reasonable upper bound for reading and writing checkpoint files is 2,000 seconds (~30 minutes) and that translates to a minimum bandwidth of 15 GB/sec. Any bandwidth below this number will constitute a major I/O bottleneck.

11.3.4.4 Shared Data

Our group does not use shared data at this time but we will explore this in the near future.

11.3.4.5 Archival Data Storage

For 2017, we expect to generate 20 times more data based on the CPU usage. The data requirement is calculated using this estimate, although we will address the issue of computation vs. I/O in subsequent sections.

11.3.4.6 Memory Required

As pointed out by 14.4.7 below, the processing unit for upcoming supercomputers will be very different than what is available today, so it is very difficult to describe the amount of data per node as the memory per node will very much depend on the type of computing hardware that resides within a node. However, we believe the following points will be true in 2017.

- Just like 2012, the single-node performance is optimal when there is a large amount of data sitting on the compute node. Therefore we expect to use a large amount of memory per core regardless of the type of computing hardware that makes up the compute node.
- The overall memory requirement will increase by two - three by 2017 (see 14.4.1).

11.3.4.7 Many-Core and/or GPU Architectures

The UCLA simulation group has been working toward optimizing our suite of codes for upcoming processors. Although it is not clear whether the next supercomputer will be many-core CPUs or GPUs, the next generation supercomputers will use SIMD processors where the main bottleneck is memory bandwidth. Our strategy is to develop a portable data structure that will maintain the flow of data to the large number of processing units in order to maintain good performance. We believe this strategy will work for all of the next-generation processors.

11.3.4.8 Software Applications and Tools

Currently our GPU-enabled codes are written in CUDA. The UCLA group is exploring various SIMD architectures, including the SSE vector, and the GPU using CUDA, CUDA Fortran, and OpenACC, so we are in a good position to program for future HPC systems. Currently our codes (both OSIRIS and QuickPIC) use parallel HDF5 for I/O and we hope this is available for future systems. Parallel post-processing tools to process the large amount of data generated by these simulations will also be needed (see 14.4.9).

11.3.4.9 HPC Services

As supercomputers become more powerful, I/O and post-processing will become an increasing bottleneck. High-performance I/O, better visualization tools, and some other approaches (see 14.4.11) will be needed to address the growing amount of data generated by exascale supercomputers.

11.3.4.10 Time to Solution and Throughput

We do not have any particular throughput constraints at this time.

11.3.4.11 Data Intensive Needs

As more lightweight processors become available, I/O and data analysis will become a larger percentage of the total workflow. Analysis and post-processing may become a more integral part of computation to minimize the amount of data generated by exascale simulations. The trade-offs between computation and I/O will be a larger issue between now and 2017.

11.3.4.12 Additional Comments

We have been very happy with the service provided by NERSC and trust NERSC will continue to provide a useful HPC platform to its users.

11.3.5 Requirements Summary Worksheet

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	8.3 M (27 M everywhere)	166 M (540 M)
Typical number of cores* used for production runs	15,000	>200,000
Maximum number of cores* that can be used for production runs	300,000	Unknown[1]
Data read and written per run	20 TB	40 TB
Minimum I/O bandwidth	~5-10GB/sec	>10GB/sec
Percent of runtime for I/O	5-10	10-15[2]
Archival data	90 TB	1,800 TB
Memory per ~~node~~ (core)	1 GB	.5 GB (minimum)
Aggregate memory	10 TB	>15 TB

- "Conventional cores."

[1] This number depends on code development and the direction of future hardwares, although we have been very successful in porting our codes to all state-of-art supercomputers in the past.

[2] This number depends on the bandwidth of future servers, but we hope this number does not exceed 10-15%.

11.4 Advanced Modeling for Particle Accelerators

Principal Investigator: Kwok Ko (SLAC National Accelerator Laboratory)
Senior Investigators: Lixin Ge, Oleksiy Kononenko, Zenghai Li, Cho Ng, Liling Xiao (SLAC National Accelerator Laboratory); Andrei Lunin (FNAL); Haipeng Wang (JLab); Sergey Belomestnykh (BNL); Ali Nassiri (ANL); Esmond Ng (LBNL); Matthias Liepe (Cornell); Jean Delayen (ODU); Frank Marhauser (Muplus)
NERSC Repository: m349

11.4.1 Summary and Scientific Objectives

Particle accelerators constitute a significant portion of DOE's investment portfolio in the Office of Science covering important facilities in the HEP, NP and BES program offices. The accelerators in operation include RHIC at BNL, CEBAF at TJNAF, SNS at ORNL, and LCLS at SLAC while those under development, being planned or proposed include the LHC upgrade, ILC, CLIC, Project X and the Muon Collider in HEP, the CEBAF Upgrade and FRIB in NP, and the LCLS-II, APS Upgrade, ERL and NGLS in BES. To build and optimize the performance of these large and expensive scientific instruments, numerical modeling and simulation have been absolutely essential to verify the design of the machine to lower cost and reduce risk. High performance computing, in particular, that utilizes the scale and speed of DOE ASCR's advanced computers, has been vital in enabling the massive amount of computations required to meet the increased complexity, accuracy, and resolution needed for the design of the next generation accelerators.

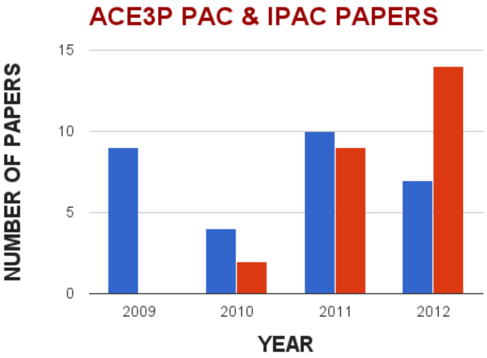
The goal of the Advanced Modeling for Particle Accelerators (AMPA) project at SLAC is to solve the challenging design and operational problems facing existing and future accelerator facilities by harnessing the compute power of DOE ASCR's flagship machines using the modeling tools developed under SciDAC and SLAC support collectively known as ACE3P. The ACE3P (Advanced Computational Electromagnetics 3P) software is a suite of 3D parallel finite-element based electromagnetic codes developed to run on massively parallel HPC platforms such as those at NERSC, to allow accelerator designers and engineers to model and simulate complex RF components and accelerator systems on a scale and at a speed previously not possible. This set of scalable codes has been vigorously benchmarked against measurements and successfully applied to a broad range of applications in accelerator science, accelerator development and program facilities worldwide. Using ACE3P, complicated accelerator components are routinely modeled to machining accuracies, hence making virtual prototyping a reality. With state-of-the-art computing resources at NERSC, even start-to-end simulation is now possible on a time scale that can impact the overall accelerator design.

The success of ACE3P led to three Code Workshops at SLAC - CW09[10], CW10[11] and CW11[12], resulting in a world-wide community of over 50 users from national labs,

[10] http://www-conf.slac.stanford.edu/CW09/default.asp
[11] http://www-conf.slac.stanford.edu/CW10/
[12] http://www-conf.slac.stanford.edu/cw11/default.asp

universities and industry and growing. Specifically they include from the Americas – 9 accelerator facilities, 6 universities and 3 companies, from Europe – 5 accelerator facilities, 2 universities, and from Asia – 4 accelerator facilities and 2 universities. Materials from CW11, including the tutorials for the ACE3P modules and the user manual, are available on line[13]. Under AMPA, the effort to broaden the ACE3P user base will continue to further advance code capabilities, foster collaborations and educate young researchers in HPC with the aim for a more productive and cost-effective program in computing for accelerator R&D. It is important that the ACE3P code suite be included in HEP's plan for Accelerator Stewardship going forward so that this core set of scalable modeling and simulation tools remains to be available to future machine designers and facility builders. Furthermore ACE3P has the potential to replace expensive commercial electromagnetic software in use by the accelerator community thus lowering the project cost for facility research and development throughout the DOE complex and beyond.

Presently AMPA focuses on supporting the HEP General Accelerator R&D (GARD) through the application of the ACE3P parallel simulation code suite in many areas. They include surface field enhancement investigations in *high gradient c*avities, power coupling designs in photonic bandgap (PBG) fibers for *dielectric laser acceleration,* the determination of mode oscillation in *klystron* cavities, as well as cavity imperfection studies in Project X *superconducting (SRF)* cavities (funded by SciDAC). AMPA also continues the *cavity design and optimization* efforts for LARP (crab cavity design for the LHC upgrade), MAP (optimization of Muon cooling cavity) and Project X (main injector cavity). All these activities are planned to continue through 2017. As a result of the Code Workshop series, the ACE3P users (CW11 attendees in photo) have contributed an increasing number of PAC and IPAC papers (SLAC papers shown in blue and papers by the user community in red) because of the continued computational support from SLAC benefitting the design and research of accelerator facilities in the US and abroad.



By 2017 AMPA expects to extend ACE3P's capabilities to three important areas - Superconducting Radio-Frequency (SRF) R&D, system-level cavity chain modeling that includes multi-physics effects, and end-to-end RF source simulation. SRF cavities are central to various future HEP accelerator projects such as the Project X CW and pulsed linacs so a self-consistent electro-mechanical optimization tool to minimize microphonics

[13] https://confluence.slac.stanford.edu/display/AdvComp/Materials+for+cw11

and/or Lorentz force detuning phenomena is essential. For accelerator systems such as the cryomodule in Project X, integrated modeling which includes RF, thermal and mechanical effects is of great interest. AMPA also plans to improve the PIC capabilities in ACE3P to enable end-to-end simulation of klystrons, which is in need of a large-scale high fidelity, high accuracy and fast modeling tool based on HPC. If successful, this high-risk high-payoff endeavor will raise the US competitiveness in "design for manufacturing" in the microwave industry potentially impacting the nation's economy.

11.4.2 Methods of Solution

Under the DOE Accelerator Grand Challenge, SciDAC-1, SciDAC-2 and SciDAC-3 programs, SLAC has developed the ACE3P (suite of parallel codes which focuses on the solutions to Maxwell's equations based on the higher-order, curvilinear finite-element method using the hierarchical H(curl) *Nedelec-type* basis functions discretized on a tetrahedral grid. The code suite presently consists of six modules: Omega3P (eigensolver), S3P (S parameter), T3P (wakefields and transients), Track3P (multipacting and dark current), Pic3P (RF guns and klystrons), and TEM3P (multiphysics effects). Together they comprise a comprehensive set of capabilities in the time and frequency domain useful for academic research as well as facility development over a wide range of acceleration applications. A brief description of the methods of solution for the most commonly used modules - Omega3P, T3P, and Track3P follows.

Applied in the frequency domain, Omega3P formulates Maxwell's equations as a linear or nonlinear eigenvalue problem and solves a system of large sparse matrices to find resonant modes in lossless and lossy accelerator cavities. The mathematical algorithms used include Exact Shift-Invert Lanczos method for real eigenvalue problems, Second-order Arnoldi method for complex quadratic eigenvalue problems, and Iterative Jacobi-Davidson method for complex nonlinear eigenvalue problems. Because the eigenvalues of interest are interior in most cases, also included are sparse direct solvers and Krylov subspace methods with spectral multilevel preconditioner for shifted linear systems.

Used in the time domain, T3P formulates Maxwell's equations as a second-order vector wave equation that is then solved via the implicit Newmark-beta scheme to simulate the transient field response of RF structures due to excitations by imposed fields and the wakefields generated by the transit of a rigid beam. At each time step, the linear system consisting of a symmetric positive definite matrix with different right hand sides is solved by the conjugate gradient method with a block Jacobian pre-conditioner.

Track3P solves the relativistic equation of motion of a particle under prescribed electromagnetic fields (generated with Omega3P, S3P or T3P) using the Boris scheme or Runge-Kutta method to simulate multipacting or dark current phenomena in RF cavities. This module contains the surface physics for primary field emission (Fowler-Nordheim law) and secondary emission (secondary emission yield or SEY curve) and is able to model multipacting and dark current realistically due to the high accuracy fields provided by the EM modules (Omega3P, S3P or T3P) using higher-order curved finite elements.

Descriptions for the remaining three modules (S3P, Pic3P and TEM3P) can be found in the CW11 tutorials.

11.4.3 HPC Requirements

The ACE3P code suite, written in C++ and using MPI, has been implemented on scalable computing platforms at NERSC. The problem size in a 3D finite element code such as ACE3P depends on the number of elements (n) on the mesh and the order of the basis function (p) for each element which together determines the matrix size N. Hence a simulation with p = 2 (2nd order), N = 6.2 * n while one with p = 3 (3rd order), N = 18 * n and p can go as high as 6th order resulting in a huge matrix.

A strong motivation for developing scalable solvers in Omega3P is the need to compute the eigenmodes in very large accelerator structures such as the multi-cavity chains in the cryomodule for Project X. In the Omega3P algebraic eigenvalue problem, one solves a series of highly indefinite linear systems and a spectral transformation is needed to obtain the interior eigenvalues, which are the ones of interest. While sparse direct solvers can be used to solve highly indefinite linear systems, they unfortunately suffer from imbalanced and non-scalable per-node memory usage. As a result the amount of available per-node memory becomes the main constraint on how large a problem size Omega3P can handle thereby limiting its parallel scaling. The development of a spectral multi-level pre-conditioner has led to an order of magnitude increase in problem size and progress has been made on a scalable hybrid solver through the collaboration with the SciDAC CETs as shown in Fig. 1.

In the T3P simulation, the run-time is proportional to the number of time steps, which in turn is determined by the element size and the highest frequency to be resolved in the structure. Using the conjugate gradient method and applying a block Jacobian pre-conditioner at each time step with each core owning one block, an incomplete factorization is performed and Fig. 2 shows that the method is very efficient and scalable.

For particle tracking in Track3P, each processor owns the whole mesh and all the field data and the particles divided evenly among all the processors. As there is no communication between processors, the computation is embarrassingly parallel with good load balancing but when the mesh and field data become too large, the memory per processor is the limiting factor.

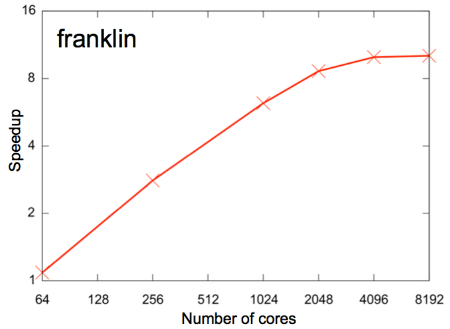


Fig. 1 Omega3P strong scaling on Franklin

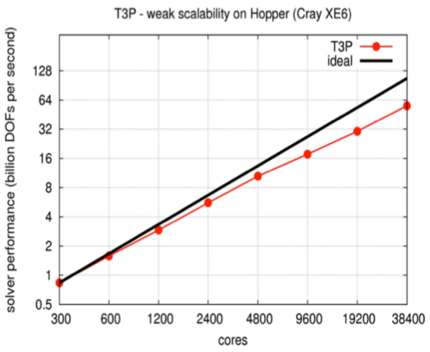


Fig. 2 T3P weak scaling on Hopper

11.4.4 Computational and Storage Requirements Summary for ACE3P

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours	3.2M	5M
Typical number of cores* used for production runs	4,800	10,000
Maximum number of cores* that can be used for production runs	40,000	100,000
Data read and written per run	1 TB	10 TB
Maximum I/O bandwidth	20 GB/sec	50 GB/sec
Percent of runtime for I/O	20	20
Shared filesystem space	5 TB	20 TB
Archival data	38 TB	50 TB
Memory per node	24 GB	64 GB
Aggregate memory	0.5 TB	2 TB

* "Conventional cores."

11.4.5 Support Services and Software

The modeling and simulation workflow for the HEP AMPA project to support the RF cavity design and optimization efforts for the accelerator community follows the sequence of model generation with CAD program, meshing with CUBIT, partitioning with ParMetis, computing/analysis with ACE3P and visualization with ParaView. Other necessary software includes LAPACK, ScaLAPACK, PETSC, MUMPS, and SUPERLU. HPSS and the NERSC Global File System are needed for data resources as well parallel NETCDF which is the IO library used for checkpointing and result data.

11.4.6 Emerging HPC Architectures and Programming Models

In preparation for the transition to many-core systems, work has started on the development of eigensolvers and linear solvers that are scalable on these emerging HPC architectures. In collaboration with the *FastMath SciDAC Institute* under SciDAC-3*,* the implementation of a hybrid solver that can reduce memory usage is in progress while code optimization effort, which can help to improve the runtime performance on these heterogeneous computing systems, is also being considered.

Appendix A. Attendee Biographies

Julian Borrill is co-lead of the Computational Cosmology Center, a Senior Staff Scientist in the Computational Research Division at Berkeley Lab and a Senior Research Physicist at the Space Sciences Laboratory at UC Berkeley. His current work is focused on developing the high performance computing tools that will be needed to analyse the data from the coming generation of cosmic microwave background polarization experiments, and applying them to the Planck satellite, Polarbear ground-based, and EBEX balloon missions. For the last decade he has also managed the CMB community and Planck-specific HPC resources at the DOE's National Energy Research Scientific Computing Center.

Richard Brower is a member of the Joint Faculty in the Departments of Physics and Electrical & Computer Engineering at Boston University. With a Ph.D. in Physics from University of California, his research interests are in non-perturbative problems in quantum field theory applied to Quantum Chromodynamics and possible new strong gauge dynamics beyond the Standard Model (BSM) for electroweak symmetry breaking at the TeV scale; methods for computational Lattice Field Theory; AdS/CFT duality and string theory; and research into new algorithmic multigrid methods to resolve multiscale physics and targent heterogeneous hardware with GPU accelerators.

Andrew Connolly is a Professor of Physics in the Dept. of Astronomy at University of Washington.

Scott Dodelson is a Scientist at Fermi National Accelerator Laboratory and Professor in the Department of Astronomy and Astrophysics and the Kavli Institute for Cosmological Physics at the University of Chicago. He received his PhD from Columbia University, after which he did post-doctoral work at Harvard University and Fermilab. He is the author of the textbook, Modern Cosmology, and over 130 scientific papers as well as editor of two other books. Dodelson has served on the Astronomy and Astrophysics Advisory Committee and numerous other local and national committees. He is a Fellow of the American Physics Society and Editor of Physics Letters B and the Journal of Astroparticle Physics.

Cameron Geddes is a staff scientist in the LOASIS program of Lawrence Berkeley National Laboratory, investigating use of laser driven plasma waves to build compact next generation particle accelerators and photon sources. These accelerators sustain much higher accelerating fields than conventional devices. Applications include extending the future reach of high energy physics and compact sources of near-monochromatic MeV photons for nuclear interrogation. Geddes received the Ph.D. in 2005 at the University of California, Berkeley, supported by the Hertz Fellowship, receiving the Hertz and APS Rosenbluth dissertation prizes for the first laser plasma accelerator producing mono-energetic beams. He received the B.A. from Swarthmore College in 1997, and the APS Apker and Swarthmore Elmore prize for thesis work on Spheromak equilibria. Previous research included Thomson scattering measurement of driven waves in inertial confinement fusion plasmas (1997-99, LLNL), wave mixing (1999, Polymath), small aspect Tokamaks (1995, Princeton/U. of Wisconsin), and nonlinear optics (1993-95).

Steven Gottlieb is Professor of Physics at Indiana University. He works in the area of elementary particle theory mostly studying the specialty of lattice QCD.

Salman Habib a member of the High Energy Physics and Mathematics and Computer Science Divisions at Argonne National Laboratory, a Senior Member of the Kavli Institute for Cosmological Physics at the University of Chicago, and a Senior Fellow in the Computation Institute, a joint collaboration between Argonne National Laboratory and the University of Chicago. His research interests cover the broad sweep of classical and quantum dynamical systems, from field theories to particles, and from the largest scales to the smallest.

Barbara Helland is Associate Director (Acting) of the Office of Advanced Scientific Computing Research at the DOE Office of Science.

Stefan Hoeche is a theoretical physicist working at SLAC. His research interests are in the field of particle physics phenomenology, in particular perturbative QCD and the construction of Monte Carlo event generators.

Thomas LeCompte is the physics coordinator for the ATLAS experiment, a 3,000-person collaboration at the Large Hadron Collider (LHC) at CERN. This experiment studies the collisions of protons at the highest energy yet achieved.

Kwok Ko is a physicist in the accelerator research division at the SLAC National Accelerator Laboratory.

Peter Nugent is a Senior Scientist in LBNL's Computational Research Division and an Adjunct Professor of Astronomy at UC Berkeley. He is the Group Leader of the Computational Cosmology Center and the Team Leader for NERSC Analytics. Peter joined NERSC after four years as a post-doc in the Lab's Physics Division to help strengthen the computational astrophysics activity at NERSC. Peter worked with Saul Perlmutter's Supernova Cosmology Project and used NERSC's supercomputers to perform thousands of supernova simulations. As the theorist in Saul's group, Peter conducted "spectrum synthesis," starting with a theory of an exploding supernova to create a theoretical spectrum and then compare that model with observed data. More recently, Peter architected and led the Deep Sky project. Deep Sky is one of the largest repositories of astronomical imaging data (over 80 TBs) and is the backbone of the Palomar Transient Factory, currently the largest source for the discovery of new astrophysical transients in the world. Deep Sky represents Peter's growing interest in data intensive science, a field that includes advanced algorithms, data management tools, storage and communication systems, along with visualization. Peter earned his Ph.D. in physics, with a concentration in astronomy, from the University of Oklahoma.

Michele Papucci is a professor of Physics at the University of Michigan.

Rob Roser is the head of the Scientific Computing Division at Fermilab.

Elizabeth S Sexton-Kennedy is the Compact Muon Solenoid (CMS) Deputy Department Head, Scientific Programs, Fermi (IL) National Accelerator Laboratory [Fermilab] In addition she is the L1 Offline Software Coordinator for international CMS.

James Siegrist is Associate Director for High Energy Physics at the DOE Office of Science.

Panagiotis Spentzouris is a scientist in the Computing Division and the Accelerator Physics Center of the Fermi National Accelerator Laboratory. Since 2001, his main research interest has been computational accelerator physics. He serves as the head of the Accelerator and Detector Simulation and Support department in the Computing Division, and is the PI of the SciDAC2 ComPASS project.

Doug Toussaint's research involves the use of massively parallel computers to calculate some of the most fundamental quantities in high-energy physics. He employs lattice gauge theory to calculate the masses and lifetimes of strongly interacting particles, the weak interactions of these particles, the behavior of nuclear matter at very high temperatures, and the structure of the electroweak interactions. Toussaint is a professor in the Physics Department at the University of Arizona. He earned his Ph.D. in Physics from Princeton University in 1978.

Frank Tsung is an Associate Professor in Physics and Astronomy at UCLA.

Craig Tull is group leader of the Science Software Systems group in the Advanced Copmputing for Science Department at Lawrence Berkeley National Laboratory. Tull has a Ph.D. in Physics from University of California, Davis, and has been developing scientific software and managing software projects for more than 25 years. His interests are in component frameworks, generative programming, and using scripting languages to enhance the power and flexibility of scientific data exploration. He has worked on science frameworks for several experiments, including as framework architect in the STAR experiment, and as leader of the LBNL framework effort in ATLAS. Tull has worked on the PPDG (Particle Physics Data Grid) and the GUPFS (Global Unified Parallel File System) projects that aim to deliver innovative solutions to data-intensive computing in the distributed environment. He recently ended a three-year assignment in DOE headquarters as program manager for Computational High Energy Physics including HEP's SciDAC portfolio, and is currently the U.S. manager of Software and Computing for the Daya Bay neutrino experiment in China.

Torre Wenaus is a Staff Physicist in the Physics Applications Software Group of the

Physics Department at Brookhaven National Laboratory.

Stan Woosley's interests in the origin of the elements and the death of massive stars have led him to do theoretical work in diverse fields. On the one hand, he studies nucleosynthetic "processes," the nuclear physics and theoretical astrophysics whereby the jigsaw puzzle of abundances that we see in stars has been assembled. This requires a firm grounding in nuclear physics, but also a thorough understanding of the lives of stars and their deaths as supernovae. Since the latter is poorly understood, Woosley and his many collaborators also use supercomputers and develop the necessary software to study supernovae and gamma-ray bursts of all types. Woosley proposed the "collapsar" model for gamma-ray bursts and was a co-investigator on the High Energy Transient Explorer that studied them. He is currently the Principal Investigator for the nine-institution Computational Astrophysics Consortium funded by DOE's Scientific Discovery through Advanced Computing (SciDAC) Program. This consortium is dedicated to a better understanding of supernovae of all types achieved through a combination of analytic studies and supercomputer models. Woosley is a professor in the Physics Department at UC Santa Cruz and has a Ph.D. in Space Science from Rice University.

Appendix B. Meeting Agenda

Time	Topic	Presenter	Science Area
8:00	Arrive, informal discussions		
8:30	Welcome & Introductions	Dave Goodwin	NERSC Program Manager
8:45	Welcome & Meeting Goals	Barbara Helland	Acting ASCR Associate Director
9:00	HEP Program Office Research Directions	James Siegrist	HEP Associate Director
9:30	NERSC Role in HEP Research & Emerging Technologies	Sudip Dosanjh	NERSC Director
10:00	Break		
10:15	Meeting outline and expectations	Harvey Wasserman	NERSC, Meeting Coordinator
10:30	Case Study: Lattice Gauge Theory Calculations	Doug Toussaint, Steven Gottlieb, Richard Brower	Energy and Intensity Frontiers: Theory
11:10	Case Study: Theoretical Particle Physics Simulations for LHC Processes	Michele Papucci & Stefan Hoeche	Energy and Intensity Frontier: Theory
11:30	Case Study: Cosmological Simulations for Sky Surveys	Salman Habib	Cosmic Frontier: Theory
	Break		
1:00	Case Study: Baryon Oscillation Spectroscopic Survey and/or Distance Supernova Search	Peter Nugent	Cosmic Frontier: Experiment
1:30	Case Study: Cosmic Microwave Background Data Analysis	Julian Borrill	Cosmic Frontier: Experiment
2:30	Case Study: The Large Synoptic Survey Telescope (LSST)	Andrew Connolly	Cosmic Frontier, Experiment
3:00	Case Study: The Dark Energy Survey (DES)	Scott Dodelson	Cosmic Frontier, Experiment
	Break		
3:30	Case Study: Detector Simulations using GEANT 4	Rob Roser and Tom LeCompte	Energy and Intensity Frontiers: Experiment
3:50	Case Study: Energy Frontier Data Analysis	Elizabeth Sexton-Kennedy, Torre Wenaus	Energy Frontier: Experiment
4:10	Case Study: Intensity Frontier Data Analysis (Daya Bay)	Craig Tull	Intensity Frontier: Experiment
4:30	Case Study: Community Petascale Project for Accelerator Science and Simulation	Panagiotis Spentzouris	Accelerator Science and Modeling
5:00	Case Study: Plasma Accelerator Simulation Using Laser and Particle Beam Drivers	Cameron Geddes, Frank Tsung	Accelerator Science and Modeling
5:30	Case Study: Advanced	Kwok Ko	Accelerator Science and

	Modeling for Particle Accelerators		Modeling
5:40	Discussion: Large Astrophysical Data Sets	Salman Habib, Moderator	
6:00	Adjourn for Day		

Friday, May 27		
8:00am	Arrive, informal discussions	
8:30	NERSC Initial Summary	Richard Gerber
9:30	Case study format review; sample case study	Harvey Wasserman
10:00	Break	
10:45	Report schedule and process	Richard Gerber
11:00	Q&A, general discussions, breakout sessions, and lunch	
1:00pm	Adjourn	

Appendix C. Abbreviations and Acronyms

ALCF	Argonne Leadership Computing Facility
AMR	Adaptive Mesh Refinement
API	Application Programming Interface
ASCR	Advanced Scientific Computing Research
ATLAS	A Toroidal LHC Apparatus (LHC particle detection experiment)
AY	Allocation Year
CDF	The Collider Detector at Fermilab
CMS	Compact Muon Solenoid (LHC particle detection experiment)
CUDA	Compute Unified Device Architecture
DES	Dark Energy Survey
DESSN	Dark Energy Survey Supernova Search
EIC	Electron Ion Collider
ESnet	DOE's Energy Sciences Network
FDTD	Finite Difference Time Domain
FEL	Free Electron Laser
FEM	Finite Element Modeling
FFT	Fast Fourier Transform
FNAL	FermiLab National Accelerator Laboratory
FRIB	Facility for Rare Isotope Beams
GCR	Generalized Collisional-Radiative
GPGPU	General Purpose Graphical Processing Unit
GPU	Graphical Processing Unit
HDF	Hierarchical Data Format
HEDP	High Energy Density Physics
HPC	high-performance computing
HPSS	High Performance Storage System
HTC	High Throughput Computing
I/O	input output
IDL	Interactive Data Language visualization software
INCITE	Innovative and Novel Computational Impact on Theory and Experiment
LANL	Los Alamos National Laboratory
LBNL	Lawrence Berkeley National Laboratory
LHC	Large Hadron Collider
LLNL	Lawrence Livermore National Laboratory
LQCD	Lattice Quantum ChromoDynamics
LSST	Large Synaptic Survey Telescope
MC	Monte Carlo
MPI	Message Passing Interface
NERSC	National Energy Research Scientific Computing Center
NetCDF	Network Common Data Format
NGF	NERSC Global Filesystem
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
OS	operating system

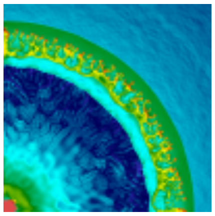
PDE	Partial Differential Equation
PDSF	NERSC’s Parallel Distributed Systems Facility
PIC	Particle In Cell
PTF	Palomar Transient Factory
RF	Radio Frequency
RHIC	Relativistic Heavy Ion Collider
SC	DOE's Office of Science
SciDAC	Scientific Discovery through Advanced Computing
SLAC	SLAC National Accelerator Laboratory
SN	Supernova

Appendix D. About the Cover

Image showing a portion of NERSC's "Hopper" system, a Cray XE6 installed during 2010. Hopper is NERSC's first peta-FLOP resource, with a peak performance of 1.28 PetaFLOPs/sec, 153,216 compute cores, 212 Terabytes of memory, and 2 Petabytes of disk. Hopper placed number five on the November 2010 Top500 Supercomputer list.

The Energy Frontier
The Intensity Frontier
The Cosmic Frontier

A Venn diagram illustrating the interlocking framework of the three interrelated frontiers of high energy physics research: energy, intensity and cosmic. HEP research probes the universe to understand fundamental particle properties, discover new phenomena and learn about the 'dark universe' through these three complementary frontiers. At the Energy Frontier, collider and fixed target experiments create new particles, reveal their interactions, and investigate fundamental forces. At the Intensity Frontier, experiments explore fundamental forces and particle interactions by studying events that rarely occur in nature. At the Cosmic Frontier, observations and measurements offer new insight and information about the nature of dark matter and dark energy.



A small portion of a visualization from a CASTRO (Eulerian Radiation Hydrodynamics) simulation of a collision between two shells of matter ejected by a massive star in two pair-instability supernova eruptions, only years apart, just before the star dies. The image shows a slice through a corner of the event. Shell radius (red knots) is about 500 times the Earth-Sun distance. Colors represent gas density (red is highest, dark blue is lowest). Image courtesy of Ke-Jung Chen, School of Physics and Astronomy, Univ. Minnesota.