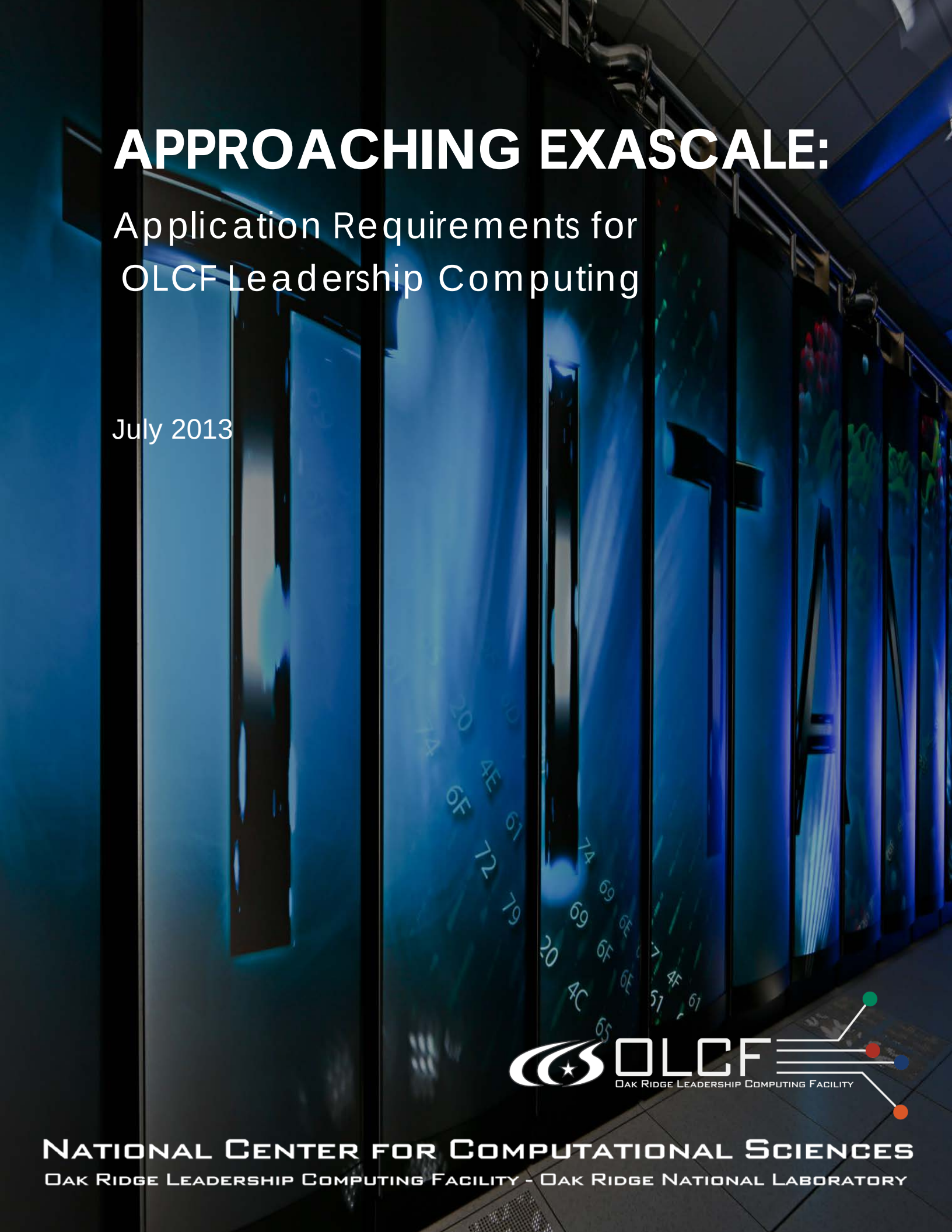


APPROACHING EXASCALE:

Application Requirements for OLCF Leadership Computing

July 2013

OLCF
Oak Ridge Leadership Computing Facility

National Center for Computational Sciences
Oak Ridge Leadership Computing Facility - Oak Ridge National Laboratory

ORNL/TM-2013/186

Oak Ridge Leadership Computing Facility
National Center for Computational Sciences

APPROACHING EXASCALE: Application Requirements for OLCF Leadership Computing

Valentine Anantharaj
Fernanda Foertter
Wayne Joubert
Jack Wells

Date Published: July 2013

Prepared by
OAK RIDGE NATIONAL LABORATORY
P.O. Box 2008
Oak Ridge, Tennessee 37831-6254
managed by
UT-Battelle, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1 Science need

The importance of high performance computing (HPC) for accelerating scientific and technological progress is widely acknowledged. The Advanced Scientific Computing Research (ASCR) program of the U.S. Department of Energy (DOE) has mandated critical research priorities for HPC in diverse science areas such as materials science, fusion energy, nuclear physics, high energy physics, nanotechnology, nuclear energy, climate science, biotechnology, and astrophysics. The continued increases in computational capabilities at the DOE Leadership Computing Facilities (LCFs) will be used to achieve such groundbreaking science discoveries as:

- Facilitating the design and certification of nuclear reactors to ensure their safety in the wake of the Fukushima nuclear accident.
- Modeling the dynamics of flame turbulence to increase fuel burning efficiency, potentially impacting the combustion processes that account for 86% of domestic energy use.
- Understanding the behavior of plasmas to support development of the ITER reactor, bringing us closer to the practical deployment of fusion energy.
- Comprehending unknown properties of our universe such as the distribution and nature of dark matter.
- Creation of a "virtual cell," for modeling a living cell in its entirety, potentially altering our understanding of the behavior and transmission of diseases.
- Understanding the behavior of new drugs, to accelerate the process of drug discovery, improve health care, and reduce costs.

1.2 OLCF leadership computing

In recent years, the Oak Ridge Leadership Computing Facility (OLCF) of Oak Ridge National Laboratory (ORNL) has deployed a series of HPC platforms of increasing magnitude in order to address pressing science needs, including a 6.4 teraflops (TF; 1 TF = 1×10^{12} floating point operations per second) Cray X1 in 2004, a 18.5 TF Cray X1e upgrade in 2005, a 26 TF Cray XT3 in 2005, a 54 TF dual-core system in 2006, an upgrade to a 119 TF Cray XT4 in 2007, an upgrade to 263 TF in 2008, a 1.375 petaflops (PF; 1 PF = 1×10^{15} flops) Cray XT5 in 2008, an upgrade to 2.3 PF in 2009, and most recently the upgrade to Titan, a 27 PF Cray XK7 system.

Attaining fundamental breakthroughs in multiple science domains will require computational capabilities on the order of an exaflop (EF; 1 EF = 1×10^{18} flops), far beyond current capabilities [1]. Pursuant to this, the OLCF has established a 10-year plan for leadership computing through 2024 (Table 1 and Figure 1), deploying capabilities that will enable computational science at the exascale and beyond [2].

Table 1. Computational science platform requirements for the OLCF

	2012	2017	2020	2024
Peak flops	10–20 PF	100–200 PF	500–2000 PF	2000–4000 PF
Memory	0.5–1 PB	5–10 PB	32–64 PB	50–100 PB
Burst storage bandwidth	NA	5 TB/s	32 TB/s	50 TB/s
Burst capacity (cache)	NA	500 TB	3 PB	5 PB
Mid-tier capacity (disk)	20 PB	100 PB	1 EB	5 EB
Bottom-tier capacity (tape)	100 PB	1 EB	10 EB	50 EB
I/O servers	400	500	600	700

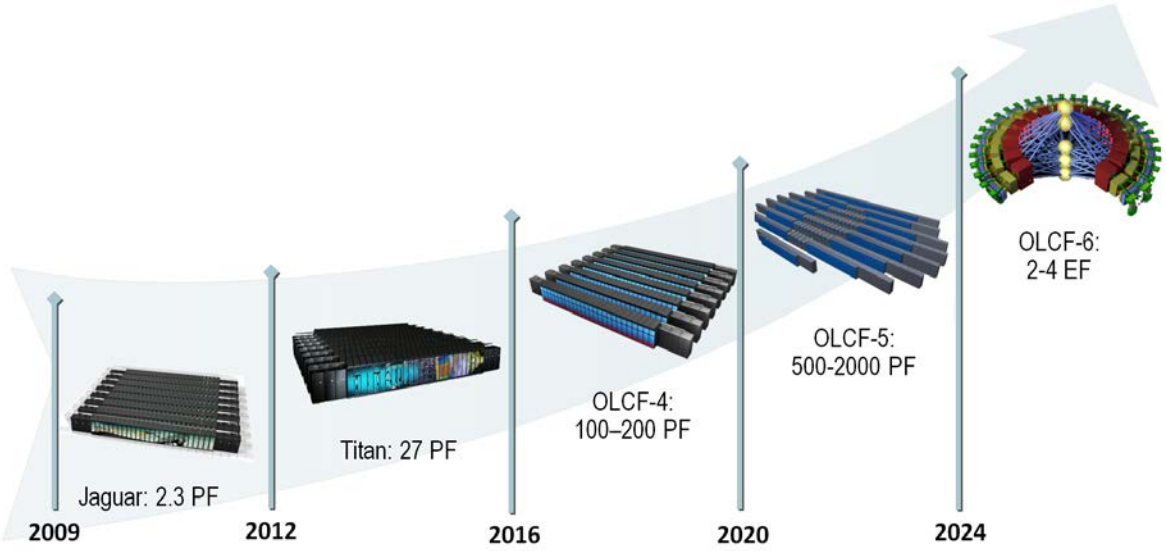


Figure 1. OLCF 2024 roadmap.

As a first step toward exascale, the OLCF began an effort in 2009 to field a next-generation computing platform of 20–30 PF compute capability based on a heterogeneous node architecture [3]. The result was Titan, a 27 PF Cray XK7 system equipped with AMD Interlagos processors and NVIDIA Kepler K20X graphics processing units (GPUs) [4]. Whereas only seven TOP500 systems employed accelerator/coprocessor technology in 2009, the number had grown to 63 by 2012 [5], and now it is generally recognized that exascale computing will require some form of compute node heterogeneity ([6]; cf. [7]). Production codes that have been ported to Titan have already undergone fundamental transitions to employing increased threading and improved memory usage, which are necessary steps in the transition to exascale.

1.3 Opportunities and challenges

Many challenges remain, however. The OLCF plans to deploy a pre-exascale system of 100–200 PF capability in the 2017 timeframe followed by an exascale system in 2020 and a 2–4 EF system in 2024. As computer hardware continues to move through a period of disruptive change, we can

expect concomitant disruptive impacts on science applications. The challenge posed by these disruptions has necessitated focused activities to prepare the scientific community, such as the DOE SciDAC Co-Design Program [8] and the DOE FastForward Program [9].

The anticipated difficulties and required technological innovations for moving to exascale are well-known [10]. These challenges are not merely in a distant future but are already being felt by both the application developer and user communities in distinctive ways:

Increasing system hardware complexity. Computer hardware is providing unprecedented levels of parallelism and performance. At the same time, system hardware is becoming more complex, diverse, and heterogeneous [11, 12], resulting in performance behaviors that can be more difficult to diagnose, understand, and exploit. Furthermore, the increasing number of parts and higher chip densities are driving the importance of resilience in ways that diverge from the needs of the broader commodity parts market [13].

Increasing software complexity. Growing hardware complexity is inducing more complexity of the system software stack as well as application software. The challenge of maintaining the reliability of an increasingly complex code base is also mirrored in the broader software community, where practices such as continuous integration are being deployed to address this concern. At the same time, science application developer teams are, in many cases, benefitting from adopting best practices being developed in the broader software developer community, often resulting in more modular and agile code bases, though there is a need and opportunity for more improvement [14].

Growing programming model complexity. The range of available programming approaches for using parallel systems has become broader, allowing more choices for a wide variety of developer needs. At the same time, the set of programming model elements commonly required to obtain good performance is complex, including the need to program for internode parallelism (MPI), intranode parallelism (OpenMP), vectorization/threading (CUDA and others), scalar performance, and in some cases processor/memory heterogeneity and resilience via checkpoint/restart. In the near future this list may grow to include staging of computations through NVRAM, user-managed fault tolerance, exploiting structured/hierarchical interconnects, energy-aware programming, and possibly programming to custom hardware. Continued movement down this path is not sustainable; system vendors and tool developers face the challenge of creating a better user-facing programming environment that is effective for science teams.

Heightened need for algorithmic innovations. As hardware characteristics become more extreme, it is increasingly difficult for applications to maintain the same fraction of peak performance at scale. New algorithmic innovations will likely be needed to stave off the collateral attrition of applications, such as task-based parallelism, out-of-core algorithms, fault-tolerant algorithms, and possibly parallelism in time. Much promising research work has already been done in these areas; future work must focus on the development of mission-ready algorithms and software to meet the growing hardware challenges.

The power challenge. Power has become an overarching problem. Predictions regarding the impact of power issues and the possible slowdown of Moore's law range from sustained progress for at least 15 more years [15] to an impending era of dark silicon [16]. OLCF and DOE leadership-

class systems are required to be power-efficient. The inability to deliver exaflop systems within the required power envelope in a timely fashion will adversely affect science teams' ability to reach the targeted science goals. Furthermore, hardware innovations in power management may very well reach into the applications space, for example, in the form of power-aware algorithms and code optimizations.

The data challenge. Multiple science domains using HPC have historically generated or used large quantities of data; by exascale, it is expected that most science domains will, and at much larger scale [17]. Improvements in the management and analysis of science simulation data offer the promise of new types of scientific discovery and insight. Large-scale data and workflow management capabilities must be developed to address this growing need.

1.4 Purpose of report

This report describes application requirements for OLCF-4, the next step on the OLCF roadmap: a pre-exascale system deployed in the 2017 timeframe. The conclusions drawn are based on the needs expressed by current OLCF users for delivering science in this timeframe. Users were surveyed on a wide range of topics, including hardware characteristics, programming model, readiness for new system features, data requirements, and workflow (see Appendix).

Requirements gathering for leadership-class systems is an inexact science. Workloads of OLCF leadership systems vary from year to year as new projects are awarded time and older projects expire. Nonetheless, the slate of projects, codes, algorithms, and science models is continuous enough to estimate, with sufficient accuracy, the needs of users within the required time period.

The remainder of this report is organized as follows. In Chapter 2 we examine the fundamental science goals driving the demands for the OLCF-4 system and beyond. In Chapter 3 we describe recent OLCF computational workloads as an indication of anticipated future system usage. Chapter 4 presents findings from the OLCF application requirements survey. To give perspective on preparedness issues for future systems, Chapter 5 provides an overview of recent efforts to migrate codes to the OLCF-3 Titan system. Finally, Chapter 6 concludes the report with a summary of findings.

2. SCIENCE DRIVERS

2.1 Science for the nation

Recognizing that high-performance computing plays an increasingly important role in addressing urgent science and technology challenges across the breadth of the federal science and technology enterprise, the President's Office of Science and Technology Policy (OSTP) established the High-End Computing Revitalization Task Force (HECRTF) in 2003 under the National Science and Technology Council (NSTC) to develop a plan for undertaking and sustaining a robust Federal high-end computing program to maintain U.S. leadership in science and technology. This plan [18] included a vision for leadership-computing systems, i.e., the leading edge, high-capability computers that will enable breakthrough science and engineering results for a select subset of challenging computational problems. These science and engineering challenges are those that have been unsolvable with lesser, high-performance computing resources. Moreover, a critical element of this plan is to enable access to leadership computing across the breadth of the federal science and technology enterprise. Implementing many of these policy recommendation into public law, the Department of Energy High-End Computing Act of 2004 (Public Law 108-423) established the Leadership Computing Facility (LCF) Program to be operated for the nation by DOE's Office of Science and directed the LCF to provide user access to leadership systems on a competitive, merit-reviewed basis to researchers in U.S. industry, institutes of higher education, national laboratories, and other federal agencies.

2.2 Science for the DOE mission

The DOE 2011 strategic plan [19] and 2012 addendum [20] call out four fundamental goals, three of these depending critically on advanced computing:

- **Goal 1:** Catalyze the timely, material and efficient transformation of the nation's energy system and secure U.S. leadership in clean energy technologies.
- **Goal 2**: Maintain a vibrant U.S. effort in science and engineering as a cornerstone of our economic prosperity with clear leadership in strategic areas.
- **Goal 3:** Enhance nuclear security through defense, nonproliferation, and environmental efforts.

With this strategic intent in mind, the DOE has identified the following priority goal and targeted outcome:

Priority Goal: "Lead Computational Sciences and High-Performance Computing"

Targeted Outcome: "Continue to develop and deploy high-performance computing hardware and software systems through exascale platforms."

The DOE continues to be the U.S. national leader as well as one of the critical few world leaders in the development of energy-efficient supercomputing to enable greater computational capabilities within the envelope of low power requirements.

The DOE strategic plan also states, "Scientific discovery feeds technology development and, conversely, technology advances enable scientists to pursue an ever more challenging set of questions. The Department of Energy strives to maintain leadership in fields where this feedback is particularly strong, including materials science research, bioenergy research, and high performance computing." Thus, leadership computing facilities are essential for the DOE to advance additional priority goals identified in its 2011 strategic plan:

- **Discover the new energy solutions we need**—Computer simulation helps ensure America's energy security by enabling researchers to improve combustion system and nuclear energy system performance, compress the design cycle for turbomachinery, improve carbon sequestration technologies, stabilize the electricity grid, and improve the efficiency of manufacturing, thereby enhancing U.S. economic competitiveness.
- **Extend our knowledge of the natural world**—Computer simulation enables breakthroughs in nuclear physics, high-energy physics, chemical and materials sciences, and climate science.
- **Deliver new technologies to advance the DOE mission**—Computer simulation enables discovery of new materials in energy and security-related systems, new bioenergy technologies, new combustion technologies, improved nuclear fission energy systems, and new fusion energy systems.
- **Enhance nuclear security**—ASCR-supported science contributes insights and research tools the National Nuclear Security Agency (NNSA) can use to ensure the safety and reliability of the nation's nuclear deterrent, a part of national strategy to safeguard America's nuclear security.

Maintaining U.S. leadership in computational science requires the best tools—including a succession of computer resources with world-class speed and capability. As is stated in the DOE Office of Science strategic plan [21], "Each of the [scientific] goals, and progress in many other areas of science, depends critically on advances in computational modeling and simulation. Crucial problems that we can only hope to address computationally require us to deliver orders of magnitude greater effective computing power than we can deploy today."

The primary mission of the DOE ASCR Program is to address these needs by discovering, developing, and deploying the computational and networking tools that enable researchers in the scientific disciplines to analyze, model, simulate, and predict complex phenomena important to the Department of Energy. The Facilities Division of ASCR has articulated a 10-year strategic plan [22] that responds to the challenge of DOE's Strategic Plan through a balanced program providing scientists and engineers across a broad range of disciplines with leadership-class computing resources while fostering the architectural development of the next generation of high-end computer hardware and supporting software.

2.3 Recent OLCF achievements

The Oak Ridge Leadership Computing Facility (OLCF) has made numerous achievements to advance these goals, including for example [23]:

- Calculated with quantified uncertainties the number of stable and radioactive isotopes in the universe that is theoretically possible under the laws of physics. Answering one of the fundamental questions of nuclear structure physics, this achievement predicted the limits of nuclear stability by determining there are approximately 7,000 possible combinations of protons and neutrons allowed in bound nuclei with up to 120 protons.
- Achieved new understanding of magnetic-reconnection physics in the Earth's magnetosphere, where it plays a key role in driving space weather. These results suggest that reconnection can spontaneously generate turbulence driving more efficient heating and transport of energetic particles into the Earth's magnetosphere, exaggerating the severity of space weather effects.
- Elucidated that carbon dioxide drove global warming at the end of the last ice age, largely bringing to a close the 40-year scientific debate on how the last glacial maximum came to an end. For the first time, an Intergovernmental Panel on Climate Change–class Coupled General Circulation Model used to predict climate's future is shown to be capable of reproducing its past. Simulations explained the lag in global surface temperature in response to carbon dioxide, and that increased insolation disrupted the Atlantic Meridonal Overturning Current (AMOC).
- Predicted that sea-level rise will continue even with the implementation of aggressive greenhouse gas mitigation policies. This project predicted, with quantified uncertainties, the amount of sea-level rise that will occur due to thermal expansion in response to mitigation strategies.
- Predicted the existence of strongly correlated electronic states in magnetic materials leading to the experimental verification of Bose Glass state of matter. This project performed a comprehensive, accurate study of the Bose glass in a doped quantum magnet within a strong magnetic field and calculated the quantitative features associated with such a state. These calculations enabled the tuning of the system between the Bose glass phase and Mott insulating phase by varying the dopant level in the magnetic material.
- Designed a turbocompressor and turbogenerator with shockwave-based technology that will lead to dramatically lower costs and higher efficiency while at the same time contributing to DOE goals for reducing carbon capture and sequestration costs. Ramgen Inc., NUMECA International, and the OLCF have transformed the workflow of this turbomachinery design project in a way that exploits the strengths of leadership computing.
- Molecular dynamics simulations reveal, and neutron-scattering experiments confirm, the self-similar, multiscale structure of lignin. In doing so, this result is a substantial step in the quest for cheaper biofuels.

The OLCF is transforming science, engineering, and technology by delivering the world's most capable computational systems to help solve the most compelling and time critical scientific and engineering problems of national importance. In the following reports, scientists from many areas of basic research, national security, nuclear energy, fusion, climate, and biology have articulated the scientific questions that may be better understood with exascale computational systems at the Leadership Computing Facility.

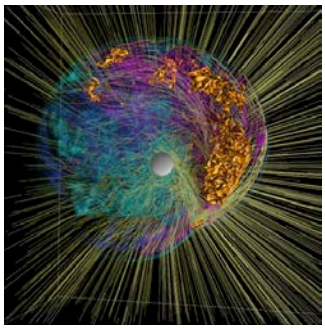
- *ASCAC Subcommittee Report: The Opportunities and Challenges of Exascale Computing (2010)* [24]

- *Discovery in Basic Energy Sciences: The Role of Computing at the Extreme Scale (2009)* [25]
- *Forefront Questions in Nuclear Science and the Role of Computing at the Extreme Scale (2009)* [26]
- *Fusion Energy Sciences and the Role of Computing at the Extreme Scale (2009)* [27]
- *Opportunities in Biology at the Extreme Scale of Computing (2009)* [28]
- *Challenges for the Understanding the Quantum Universe and the Role of Computing at the Extreme Scale (2009)* [29]
- *Challenges in Climate Change Science and the Role of Computing at the Extreme Scale (2008)* [30]
- *Science Based Nuclear Energy Systems Enabled by Advanced Modeling and Simulation at the Extreme Scale (2009)* [31]
- *Scientific Grand Challenges in National Security: The Role of Computing at the Extreme Scale (2009)* [32]
- *Exascale Workshop Panel Meeting Report (Trivelpiece 2010)* [33]

Taking a refreshed, updated look into the future, numerous science areas have identified the need for computational resources several orders of magnitude beyond what is currently available. Specific science goals are discussed below.

2.4 Science drivers

2.4.1 Astrophysics



High-end scientific simulation can provide answers regarding how supernovae occur, what happens when black holes merge and what is the nature of dark matter. Astrophysics research addresses physical phenomena from the smallest subatomic particles to the largest galaxies, such as the formation of elements, supernova behaviors, black holes, gravitational radiation, star formation, and dark matter. Supernova occurrences are the most spectacular events in the universe and are fundamental to element formation.

OLCF today

Increase physical fidelity of the nuclear burning module to effectively confront observations of SNe remnants and answer questions about galactic chemical evolution.

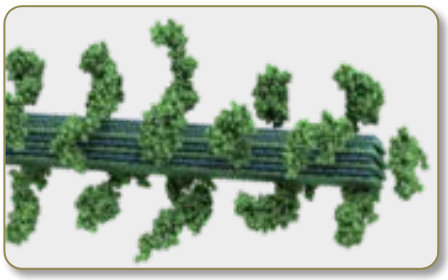
OLCF 2016

Use 3D simulations, which directly include physical couplings, to understand how the core-collapse mechanism behaves for various initial conditions, known from stellar evolution and observation.

OLCF 2022

Determine the precise manner in which supernovae explode by quantum kinetics on macroscopic scales with realistic nuclear physics components to predict isotopic output.

2.4.2 Biomass to biofuels



Enhance the understanding and production of biofuels for transportation and other bioproducts from biomass. The main challenge to overcome is the recalcitrance of biomass (cellulosic materials) to hydrolysis. Lignin itself is a very large, very complex molecule made up of hydrogen, oxygen, and carbon. In the wild, its ability to protect cellulose from attack helps plants be hardy, living in a wide range of environments. But when such biomass is used to produce biofuels, lignin is so effective that even harsh, expensive pretreatments fail to neutralize it. Nature has evolved a very sophisticated mechanism to protect plants against enzymatic attack. The goal is to understand the physical basis of biomass recalcitrance—resistance of the plants against enzymatic degradation—and engineer processes to overcome it.

OLCF today

Capabilities include atomic-detail dynamical models of biomass systems of several million atoms, permitting detailed analysis of interactions.

OLCF 2016

Perform simulations of pretreatment effects on multi-component biomass systems of tens of millions of atoms to understand the bottlenecks in bioconversion of lignocellulosic biomass to biofuels.

OLCF 2020

Simulate the interface and interaction between 100-million-atom microbial systems and cellulosic biomass, understanding the dynamics of enzymatic reactions on biomass. Design superior enzymes for conversion of biomass.

OLCF 2024

The design, from first principles, of enzymes and plants optimized for the conversion of biomass to biofuels to relieve our dependence on oil and for the production of other useful bioproducts.

2.4.3 Climate change science

Understand the dynamic ecological and chemical evolution of the climate system with uncertainty quantification of impacts on regional and decadal scales. Concerns regarding global warming and anthropogenic climate change drive the need to improve the scientific basis for assessing the potential ecological, economic and social impacts of climate change. More accurate climate models can simulate different scenarios of possible future climate change to help policy makers in their planning process.

OLCF today

Evaluate continental-scale temperature and hydrologic responses to climate forcing; models support the resolution of eddies in ocean circulation and include terrestrial biogeochemistry.

OLCF 2016

Regional climate projection on decadal time scales, including comprehensive hydrologic cycle representation, aerosol chemistry roles, and regional predictions of carbon cycle processes.

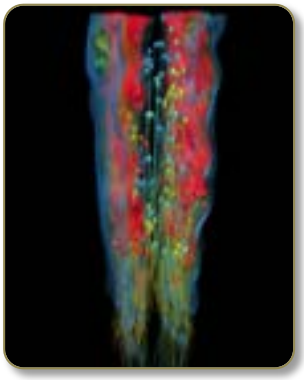
OLCF 2020

Simulate a comprehensive description of the global carbon cycle, including associated biogeochemical components (e.g., nitrogen, phosphorous). Develop higher resolution global models to support the direct numerical simulation of cloud systems. Incorporate framework to better quantify structural and parametric uncertainties.

OLCF 2024

Validated models enable understanding of the options for adapting to and for mitigating climate change on regional space scales for an arbitrary range of emissions scenarios. Fully integrate human dimensions components to allow exploration of socioeconomic consequences of adaptation and mitigation strategies. Quantify uncertainties regarding the deployment of adaptation and mitigation solutions.

2.4.4 Combustion science



Increase efficiency by 25%-50% and lower emissions from internal combustion engines using advanced fuels and new, low-temperature combustion concepts. Understanding and predicting turbulent combustion and multiphase flow phenomena in state-of-the-art transportation, propulsion, and power systems is a critical area of research for the design of advanced combustion systems. Substantial improvements in the efficiency and emissions characteristics of advanced devices are possible, but the processes are sensitive and require high levels of precision that can only be reached through the development of advanced simulation capabilities.

OLCF today

Perform 3D direct numerical simulation of turbulent combustion with small oxygenated hydrocarbons (e.g., ethanol and di-methyl ether) at low to moderate Reynolds number [34,35].

OLCF 2016

Large eddy simulation of turbulence chemistry interactions, multiphase flows, and sprays in companion benchmark optical and canonical experiments designed to provide pertinent data for model validation.

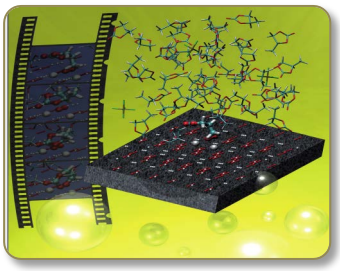
OLCF 2020

High-fidelity simulation of engine combustion in device-relevant conditions for detailed model development and reduction aimed at technical barriers associated with the design of internal combustion engines.

OLCF 2024

High fidelity simulations of engine combustion appropriate for both exploration and design incorporating the newer and deeper understanding of the fundamental physical and chemical processes in advanced combustion technologies.

2.4.5 Energy storage



Predictive engineering of safe, large format, rechargeable batteries requires an improved understanding and control of complex processes that lead to thermal runaway and fires. Widespread use of electricity generated from intermittent, renewable sources (e.g., solar, wind) and transitioning transportation to hybrid and eventually all-electric vehicles will have a dramatic effect on both oil consumption and greenhouse gas emissions and will require efficient energy storage. Although batteries and capacitors have been available for many decades, there remain many fundamental issues in understanding reaction processes at the atomic and molecular level that govern their operation, performance limitations, and failure.

OLCF today

Computational screening of thousands of candidate materials simulating ionic transport and thermodynamic properties along with limited simulations of electrochemical processes at interfaces.

OLCF 2016

Utilize inverse methods for materials design to enable the discovery of specific materials with higher energy and power densities, better stability and safety, and longer lifetimes, while addressing the processes at the interfaces. The results will have broad impact on current Li-ion technologies and enable significant exploration of beyond Li-ion solutions such as metal-air, lithium-sulfur, and multivalent systems.

OLCF 2020

Multiscale, atoms-to-devices, science-based predictive simulations of cell performance characteristics, safety, cost, and lifetime for various energy storage solutions, along with design optimizations at all battery hierarchies (battery materials, cell, pack, etc.).

OLCF 2024

Enable the development of integrated, multiscale methods for evaluating safety scenarios in real-life operation of various devices with quantified uncertainties.

2.4.6 Fusion energy / ITER

Effectively model and control the flow of plasma and energy in a fusion reactor, scaling up to ITER-size. Develop predictive understanding of plasma properties, dynamics, and interactions with surrounding materials. Fusion energy has great promise as an energy source and is one of the most difficult scientific and

engineering challenges ever attempted. Success depends on the ability to heat and electromagnetically confine the reactive plasma within the fusion reactor for a sufficient period of time.

OLCF today

Perform high-fidelity simulation of tokamak edge plasma turbulent transport for first-principles Reynolds numbers to address JET-scale plasmas with a goal of understanding high-confinement physics.

OLCF 2016

Increase simulation of tokamak edge plasma to ITER scale. Perform coupled simulations of plasma edge with core and chamber wall interactions. Control edge-localized modes and other destructive mechanisms.

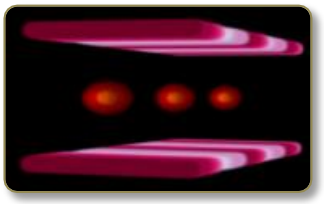
OLCF 2020

Perform integrated first-principles simulation including all the important multiscale physical processes to study fusion-reacting plasmas in realistic magnetic confinement geometries.

OLCF 2024

Produce an experimentally validated simulation capability for ITER-scale plasmas that can be used as the design and the physics study tool for DEMO, the anticipated follow-on facility, to solve the engineering issues necessary for practical electricity production with fusion plasmas.

2.4.7 Globally optimized accelerator design



Simulate experiments to validate design concepts of the Plasma Wake Field Accelerator (PWFA) Linear Collider (LC) that would have a 1000 times higher accelerating gradient. As the next generations of accelerators are planned and designed, accelerator physicists have turned to increasingly detailed numerical models. These models provide a proof of principle and a cost-effective method to design new lights sources and deploy a series of small efficiencies and dramatic new approaches. Computational simulations at OLCF are being used to study plasma-wakefield-based accelerators for experimental groups from around the world, using electron as well as proton beams.

OLCF today

Perform high-fidelity simulations of one or two key design features of PWFA-LC. Perform optimization of individual design parameters.

OLCF 2016

Begin integrating 3D simulations and begin the first stages of integrated simulations for guiding, focusing, and accelerating fields and assessing stability.

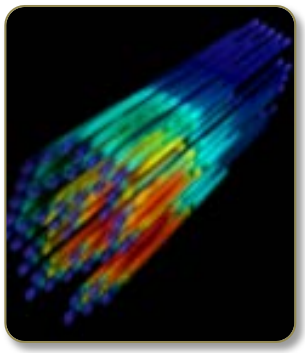
OLCF 2020

Perform global optimization of design parameters. Simulate ultra-high gradient laser wakefield and plasma wakefield accelerator structures.

OLCF 2024

Deploy virtual accelerator modeling environment for the realistic, inclusive simulation of most relevant beam dynamic effects.

2.4.8 Nuclear energy



Simulations will allow safe increased fuel utilization, power upgrades, reactor lifetime extensions, and design of new, safe, cost-effective reactors. Over the last several years, the energy security of the United States has risen in importance both politically and economically. Our nation needs to increase energy security, reduce dependence on unreliable sources of energy, obtain energy at affordable prices, and insure that the environment is not impacted. Improving scientific understanding of the behaviors of nuclear fuels, reactors, separation processes and long-term waste management sites will increase the viability of nuclear energy strategies for addressing these concerns.

OLCF today

Model 3D full-core reactor neutron transport. Predict behavior of existing and novel nuclear fuels and reactor nominal operation.

OLCF 2016

Model neutron transport-hydraulics coupling, accident scenarios and reactor transients, power ramps and accidents. Develop integrated performance and safety codes with improved uncertainty quantification.

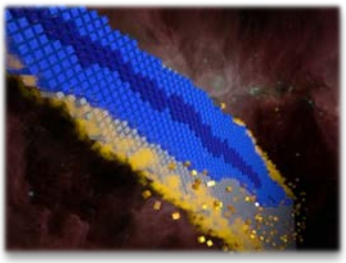
OLCF 2020

Develop integrated performance and safety codes with improved uncertainty quantification and bridging of time and length scales. Implement next-generation multiphysics, multiscale models. Perform accurate full reactor core calculations with 40,000 fuel pins and 100 axial regions.

OLCF 2024

Model multi-component chemical reacting solutions and 3D plant design, enabling the development of integrated performance and safety capabilities with improved uncertainty quantification.

2.4.9 Nuclear physics



Achieve fundamentally new insights into the nuclear many-body problem and the nature of the strong nuclear force. A detailed understanding of the atomic nucleus is both fundam entally important and of great practical significance. Not only of importance to explaining the birth of the universe and astrophysical phenomena, understanding nuclei is crucial in energy generation as well as industrial and medical applications. Nuclear physics focuses

on predicting and explaining rich classes of phenomena that occur in nuclei. The theoretical goal of increased predictive power for nuclear processes that occur in nature or in nuclear reactors, but cannot be measured in the laboratory with sufficient precision, drives the field to achieve detailed simulations using extreme-scale computers and cutting-edge algorithms.

OLCF today

High precision *ab-initio* calculations for light ion reactions.

OLCF 2016

Ab initio calculations of exotic and neutron-rich nuclei, including regions around ^{78}Ni and ^{132}Sn, that provide critical inputs to r-process nucleosynethesis.

OLCF 2020

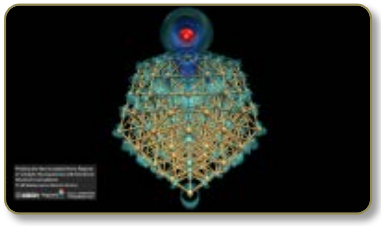
Calculations of the transport properties of neutron star crusts.

OLCF 2024

Nuclear fission calculations using *ab-initio* techniques.

2.4.10 Rational design and synthesis of multifunctional catalysts

Develop the fundamental understanding needed to design new multifunctional catalysts with unprecedented control over the transformation of complex feedstocks into useful, clean energy sources and high-value products. Computing with large-scale, high-throughput methods will play a central role because statistical mechanical sampling and free energies are fundamental concepts of this science.



OLCF today

Model electronic structures and reaction kinetics on catalytic surfaces, including homogeneous as well as nano- and meso-structured materials.

OLCF 2016

Perform computational screening of thousands of candidate materials based on databases of accurate elementary reaction rates to guide laboratory-scale system calibration. Utilize multi-scale, multi-physics methods to describe catalyst structures and reactions accurately over the necessarily long-time scales.

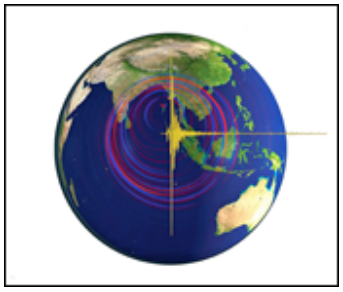
OLCF 2020

Enable end-to-end, system-level descriptions of multifunctional catalysis. Uncertainty quantification and data-integration approaches will enable inverse problems for catalytic materials design.

OLCF 2024

Enable integration of accurate, multi-scale simulations into industrial, process-level descriptions of energy production and manufacturing.

2.4.11 Seismology



Recent advances in high-performance computing and numerical techniques have facilitated fully 3D simulations of global and regional seismic wave propagation at unprecedented resolution and accuracy. These methods have advanced to the point that they may be integrated with experimental data from seismic detectors to generate global seismic images. The goal is to use the differences between observed and predicted detector signals to improve models of the Earth's subsurface and kinematic representations of earthquakes. This approach to seismic tomography is unique and is, perhaps, the first attempt to characterize the earth's makeup integrating full-wave simulations on high-performance computers with observations from such a large number of earthquakes. Scientifically, this enables fundamental physical questions such as what is the composition of the earth's interior or what is the water content of the mantle. A more accurate global model will enable better understanding of geodynamical processes such as variations in isotropic wave speed.

OLCF today

Generate a predictive model of the earth's interior using global seismic imaging and adjoint tomography using only ~250 earthquakes and a limited resolution of the time period for the observed signals (9 s).

OLCF 2016

Perform accurate global seismic imaging, assimilating data from thousands of instruments deployed across the globe, considering ~5,000 earthquakes. Generation of regional, e.g., California state-wide, physics-based probabilistic seismic hazard map at maximum resolution frequencies of 1-2 Hz; such a map at maximum frequency of 1 Hz will require 662 millions of allocation hours, going 2-hz will required 16x more allocation hours. Perform ground motion simulation up to 10 Hz.

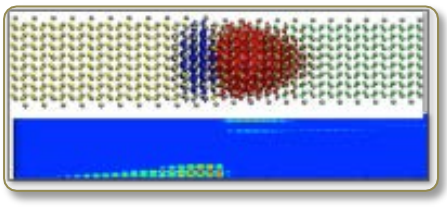
OLCF 2020

Assimilate data from more than 6,000 earthquakes. Forecast the frequency of damaging earthquakes in California over a specified time span. The inversions are then combined to provide a comprehensive view of earthquake risk in the region.

OLCF 2024

Predictive models, leading to insight into the attenuation and anisotropy of the earth's interior. We expect this will lead to a better understanding of both the physical and chemical processes deep within the earth's interior, which is not directly measurable.

2.4.12 Solar energy



Improve photovoltaic efficiency and lower cost for organic and inorganic materials. A photovoltaic material poses difficult challenges in the prediction of morphology, excited state phenomena, exciton relaxation, recombination and transport, and materials aging. The problems are

exacerbated by the important role of materials defects, aging, and complex interface morphology.

OLCF today

Perform *ab-initio* simulations of structure, carrier transport, and defect states in organic and inorganic nanosystems. Excited state phenomena can be addressed in homogeneous systems, but uncertainties are not well quantified.

OLCF 2016

Robust predictions of excited-state phenomena, including multiple exciton generation, are enabled by "beyond-DFT" methods. Excited-state behavior coupled with charge-carrier relaxation, transport, and recombination will enable prediction and understanding of growth, interface structure, and stability of heterogeneous nanophase materials and blends necessary for efficient solar conversion.

OLCF 2020

Enable computational screening of materials for desirable excited-state and charge transport properties. Systems-level, multiphysics simulations of practical photovoltaic devices are enabled.

OLCF 2024

Uncertainty quantification enabled for critical integrated materials properties (e.g., excited state dynamics, carrier transport, and relaxation). Evaluation of performance characteristics (conversion efficiency, cell lifetime) becomes possible for cell components and integrated systems.

2.4.13 Stabilizing the energy grid with dynamic power sources

Enable system-level simulation of regional electricity grids. The U.S. power grid is a huge, complex, interconnected system and is under incredible stress as energy demand exceeds the capability of the transmission system. Congestion in the grid is a large source of cost. Additionally, the existing grid is not structured to handle alternate energy sources or variable ones seen from renewable energy.

OLCF today

Develop and test new algorithms and system models capable of managing the complexity of the data and problem; experiments on the state level.

OLCF 2016

Deploy and test new algorithms and system models, expand the time horizon from more 24 hours to a more realistic 72 hours, and increase the spatial network beyond a state level.

OLCF 2020

Solve the optimization of stabilizing the energy grid while introducing renewable energy sources; incorporate more realistic decisions based on available energy sources.

OLCF 2024

Incorporate nonlinear feedback of consumers and energy suppliers to deliver more reliable energy.

3. OLCF SCIENCE WORKLOAD

3.1 Introduction

The Oak Ridge Leadership Computing Facility awards computing allocations based on three programs [36]: the INCITE (Innovative and Novel Computational Impact on Theory and Experiment) program, which enables high-impact grand-challenge research; the ALCC (ASCR Leadership Computing Challenge) program, awarding computer time to high-risk, high-payoff simulations in special situations related to DOE mission needs; and Director's Discretionary projects, a seed time allocation program primarily for enabling new and experimental efforts. The OLCF focuses on solving very large problems that cannot be solved anywhere else. As such, OLCF systems do not have fixed, rigid workloads over their lifetimes but rather have workloads that change from year to year as the mix of projects changes across a diverse range of science domains. To gauge the characteristics required for future systems, it is important to combine historical data on how OLCF systems are used with projections from current projects regarding future requirements, keeping in mind that systems must be well-balanced and general purpose in nature to allow for the contingency of new types of codes, algorithms, and use cases being deployed on our systems.

In this chapter we examine two aspects of the OLCF science workload. First we examine recent system usage characteristics primarily during calendar year 2012, covering the transition from Jaguar to Titan. We then take a more in-depth look at OLCF science applications and how they are used.

3.2 System usage characteristics

OLCF leadership systems are multipurpose in nature and serve the needs of diverse user communities. Table 2 shows the science domain categories used for this analysis and their constituent research areas.

Figure 2 shows the core-hour usage of Jaguar and Titan during 2012 by science domain. Over this period, 1.27 billion core-hours were used by 157 projects with a total of 1,502 users. Leadership system resources are used heavily in a wide variety of science domains, each with specific system requirements that must be met.

Figure 3 shows the number of OLCF projects per science domain for this period. It is seen that some science domains have particularly large numbers of small projects, such as computer science, with projects to develop next-generation tools; and engineering, with many industrial partnership projects working on extending the scope of their simulations. Likewise, Figure 4 shows the number of users by science domain. Certain domains have a large number of users, such as earth sciences which includes a large community of climate research users. It is important to note that each science domain brings together not only varied scientific applications with different algorithms, each with its own unique hardware requirements, but also highly varied developer communities with different code development practices, tool requirements, investment-heavy legacy code bases,

and science discovery workflows. The OLCF and vendors used by the center must have a good understanding of these needs in order to address the wide set of requirements of these communities.

Table 2. Research areas and science domains

Science Category	Represented Research Areas
Biology	Bioinformatics Biophysics Life Sciences Medical Science Neuroscience Proteomics Systems Biology
Chemistry	Chemistry Physical Chemistry
Computer Science	Computer Science
Earth Science	Climate Geosciences
Engineering	Aerodynamics Bioenergy Combustion Turbulence
Fusion	Fusion Energy (Plasma Physics)
Materials	Materials Science Nanoelectronics Nanomechanics Nanophotonics Nanoscience
Nuclear Energy	Nuclear Fission Nuclear Fuel Cycle
Physics	Accelerator Physics Astrophysics Atomic/Molecular Physics Condensed Matter Physics High Energy Physics Lattice Gauge Theory Nuclear Physics Solar/Space Physics

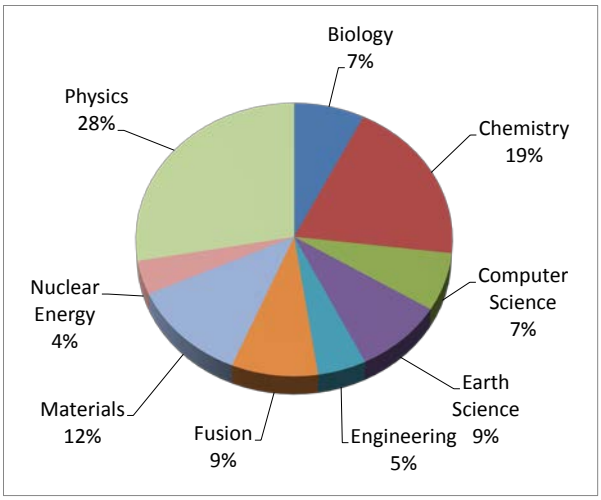


Figure 2. System usage by science domain in 2012.

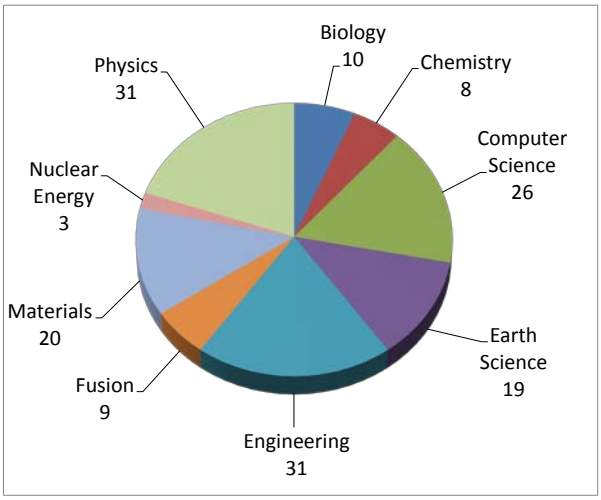


Figure 3. Number of projects by science domain in 2012.

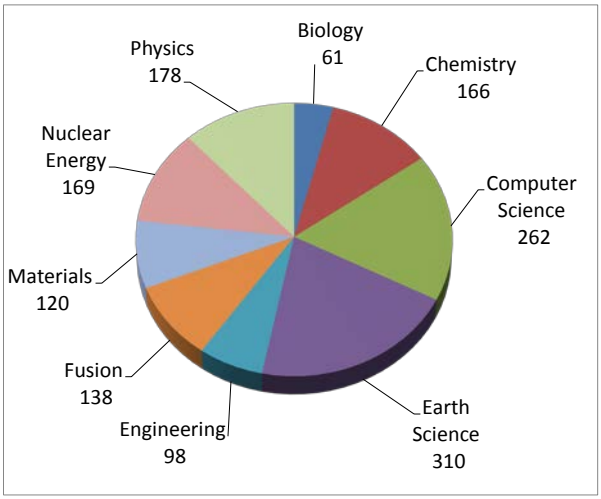


Figure 4. Number of users by science domain in 2012.

As a leadership computing center, the OLCF has a specific leadership usage metric that at least 40% of the utilized system core-hours be dedicated to jobs requiring at least 20% of the system size—for the system in 2012, roughly 60,000 cores. These values are given in Figure 5, for the nine-month period prior to a system upgrade in fall 2012. This figure shows that user jobs substantially exceed the leadership usage metric over the period. For the period in aggregate, a total of 51% of core-hours was used for jobs requiring at least 20% of the system. This can be compared to the 43% of core-hours used for leadership jobs in the period November 2009 through September 2011, during which Jaguar had 25% fewer cores. Thus, user job size requirements are outpacing the anticipated demand for leadership-sized job execution for OLCF-3.

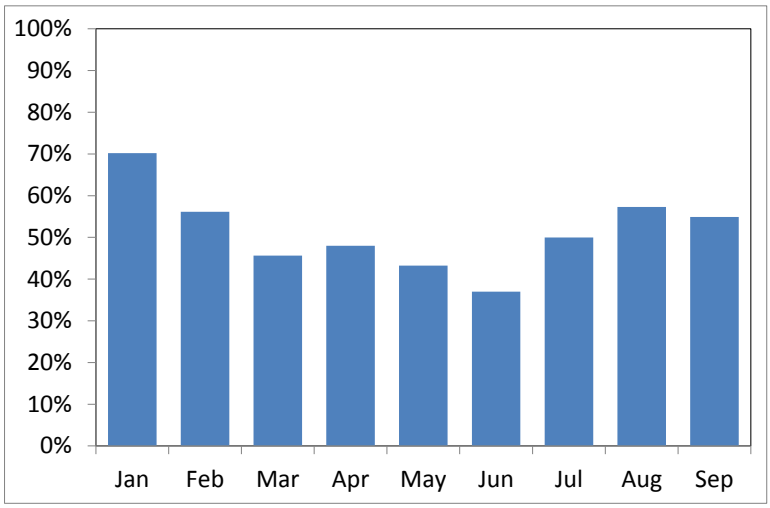


Figure 5. System leadership usage in 2012.

For formulating future requirements it is also important to understand the software components employed by users for building applications. Table 3 shows compiler types used on Jaguar and Titan for the period January 2012 through March 2013. These statistics were gathered by use of the ALTD automatic library tracking database tool [37]. The statistics show the number of users of each compiler family and the number of instances of linking an executable with each compiler type over the period. The figures do not include usage of the Nvidia NVCC compiler, which is not yet integrated into ALTD. The figures show that PGI is the most heavily used compiler family, being the default, followed by GNU. The Cray compiler is a newer product with growing usage.

Table 3. Compiler usage

Compiler family	Users	Link instances
PGI	269	347,771
GNU	168	61,865
Intel	79	17,563
Cray	62	15,323

Table 4 shows the top libraries used for this period, based on ALTD statistics. The most heavily used libraries include the Cray MPI library, the Cray LIBSCI scientific library, the CUDA toolkit, HDF5 and NetCDF I/O libraries, the FFTW library, and the PAPI profiling library. The CUDA tools are becoming more heavily used as the GPUs have become more generally available. It is clear from these figures that a small number of libraries are in extremely heavy use and thus must be well-supported and well-optimized for the targeted system.

Table 4. Library usage

Top libraries by users	Users	Top libraries by link instances	Link instances
mpt/5.5.5	387	mpt/5.5.5	410,884
libsci/11.1.01	182	cudatoolkit/5.0.35.102	53,132
libsci/12.0.00	181	libsci/12.0.00	49,578
cudatoolkit/5.0.35.102	129	libsci/11.1.01	24,752
hdf5/1.8.8	67	mpt/5.6.3	24,735
fftw/3.3.0.1	64	hdf5/1.8.8	13,677
papi/5.0.1	53	hdf5/1.8.9	13,353
hdf5-parallel/1.8.8	45	netcdf/3.6.2	12,025
mpt/5.6.3	41	papi/5.0.1	8,117
netcdf/4.2.0	40	mpt/5.6.1	7,346

3.3 Application workload characteristics

Analyzing HPC system workloads from an application-centric standpoint gives insights that are not generally available from aggregate studies of HPC job workloads. The following results cover the period November 2009 through September 2011 for the OLCF Jaguar system [38]. We consider several measures of typical OLCF application workload, such as number of applications, job size, and job duration.

HPC centers differ significantly in the breadth and number of applications run on their systems. Figure 6 is a cumulative graph of core-hours expended on Jaguar as a function of application, ranked starting with most heavily used applications. The inset graph shows that 50% of the core-hours used are spent on the top 20 applications, while the top 50 applications account for 80% of the resources used. The number of heavily used codes is small compared to some other sites that field general-purpose systems, though the figure is larger than for some special-purpose systems that support domain-specific capacity computing with a very small set of codes. The comparatively narrow range of applications used on OLCF systems enables more cogent understanding of code performance characteristics as they dictate system requirements.

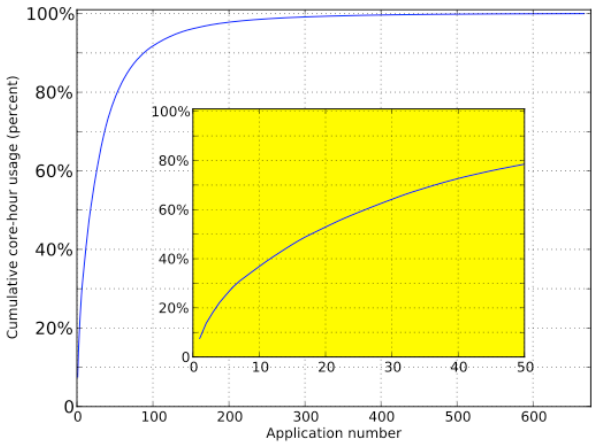


Figure 6. Jaguar core-hour usage by application.

Figure 7 is a cumulative core-hour usage graph showing the distribution of job sizes for which system core-hours are expended. OLCF systems attain the leadership metric of using at least 40% of the core-hours on jobs run at a size of 20% or more of the full system. Clearly, leadership-scale systems must be capable of running multiple science applications scalably to a large portion of the system. However, system usage is also well-distributed across the entire range of job sizes, due to the varying needs of projects and codes. OLCF systems must support this kind of job mix.

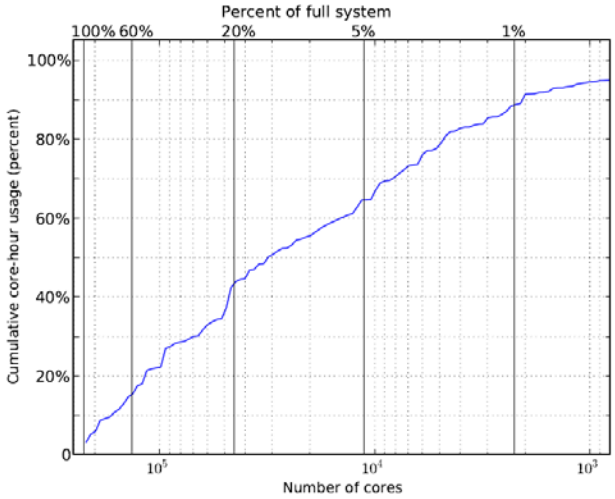


Figure 7. Jaguar core-hour usage by job size.

The usage of core-hours represented cumulatively in terms of job duration is shown in Figure 8. Jobs are normally limited to 24 hour duration by the OLCF scheduling policy. A total of 88% of core-hours was spent on jobs of 12 hours or less, and 50% of core-hours were spent on jobs of 6 hours or less. This should be compared to a system-wide mean time to failure (MTTF) of 65 hours for the period, and an average system or node failure every 35 hours. Users are effectively running their jobs within constraints of scheduler time limits and substantially below current system failure rates, in many cases using checkpoint/restart. This suggests that user applications have enough headroom to avoid resiliency problems in the short term, though the hardware resiliency situation is expected to become more challenging going forward as the number of parts increases.

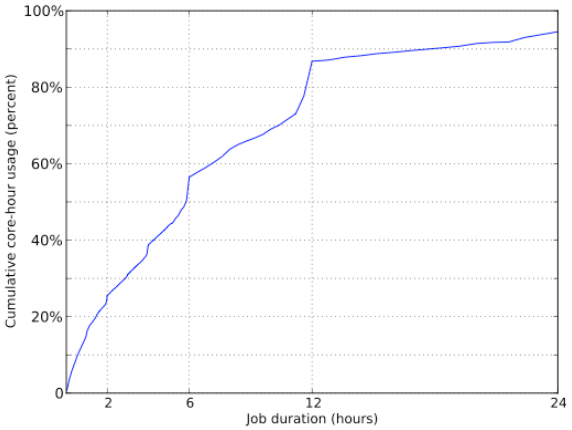


Figure 8. Jaguar core-hour usage by job duration.

Table 5 lists selected Jaguar applications that account for roughly 50% of Jaguar usage over the November 2009 through September 2011 reporting period. These include the most heavily used codes as well as several other codes of strategic interest for the future. Algorithm usage based on the algorithm motifs of each of these applications is shown in Table 6 [39]. The applications are extremely varied both in science domain and also in types of algorithms used. As leadership systems for multidisciplinary computing make the challenging move to exascale, they must continue to support the multiple hardware performance criteria necessary to run a diverse range of algorithms effectively.

Table 5. Jaguar selected applications

Application	Primary science domain	Description
NWCHEM	Chemistry	Large scale molecular simulations
S3D	Combustion	Direct numerical simulation of turbulent combustion
XGC	Fusion Energy	Particle-in-cell modeling of tokamak fusion plasmas
CCSM	Climate Research	Climate system modeling
CASINO	Condensed Matter Physics	Quantum Monte Carlo electronic structure calculations
VPIC	Fusion Energy	3D relativistic, electromagnetic, particle-in-cell simulation
VASP	Materials	Ab-initio quantum mechanical molecular dynamics
MFDn	Nuclear Physics	A Many Fermion Dynamics code
LSMS	Materials	Wang-Landau electronic structure c multiple scattering
GenASiS	Astrophysics	AMR neutrino radiation magneto-hydrodynamics
MADNESS	Chemistry	Adaptive multi-resolution simulation by multi-wavelet bases
GTC	Fusion Energy	Gyrokinetic toroidal momentum and electron heat transport
OMEN	Nanoelectronics	Multidimensional quantum transport solver
Denovo	Nuclear Energy	3D discrete ordinates radiation transport
CP2K	Chemistry	Atomistic and molecular simulations
CHIMERA	Astrophysics	Modeling the evolution of core collapse supernovae
DCA++	Materials	Many-body problem solver with quantum Monte Carlo
LAMMPS	Chemistry	Molecular dynamics simulation
DNS	Fluids and Turbulence	Direct numerical simulation for fluids and turbulence
PFLOTRAN	Geological Sciences	Multiphase, multicomponent reactive flow and transport
CAM	Climate Research	Global atmosphere models
QMCPACK	Materials	Diffusive quantum Monte Carlo simulations

Table 6. Algorithm motifs for selected applications

Application	Structured grids	Un-structured grids	FFT	Dense linear algebra	Sparse linear algebra	Particles	Monte Carlo
NWCHEM			●	●			
S3D	●			●	●	●	
XGC		●				●	
CCSM	●		●		●		
CASINO							●
VPIC	●					●	●
VASP			●	●			
MFDn					●		
LSMS				●			●
GenASiS		●			●		
MADNESS		●	●	●			
GTC	●				●	●	●
OMEN	●				●		
Denovo	●			●	●	●	●
CP2K	●				●	●	
CHIMERA	●			●	●	●	
DCA++				●			●
LAMMPS	●		●			●	
DNS	●			●	●	●	
PFLOTRAN	●	●		●	●		●
CAM	●		●	●	●	●	
QMCPACK						●	●
TOTAL	**12**	**4**	**6**	**11**	**12**	**11**	**8**

The core-hour usage of these applications is given in Figure 9, and usage characteristics for different scaling regimes are shown in Figure 10. For the latter figure, scalability is represented by showing the fraction of core-hours for each application spent in several job size brackets, including <1% of total system size, 1–5%, 5–20%, 20–60%, and >60%. These values show how users employ their core-hour allocations during production in terms of balancing usage needs and throughput. Applications with large amounts of red and especially orange color spend large parts of their allocations on leadership-scale computing, whereas large bands of blue, green, or purple indicate most core-hours are used at lower core counts. The usage patterns here are diverse, with multiple codes scaling to a large fraction of the system and others less so.

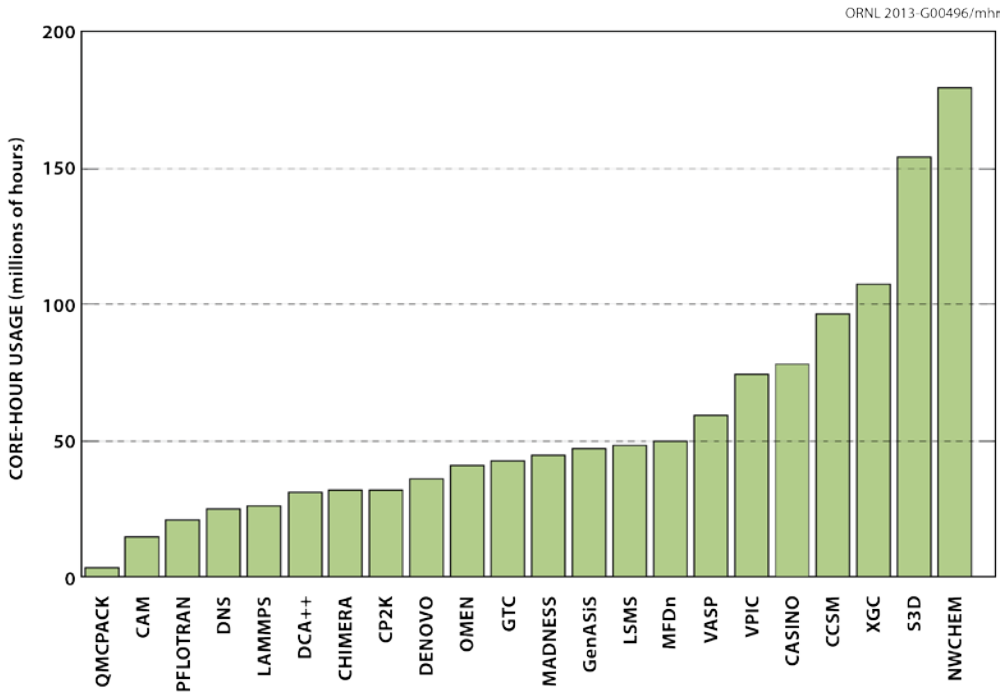


Figure 9. Core-hour usage of selected applications.

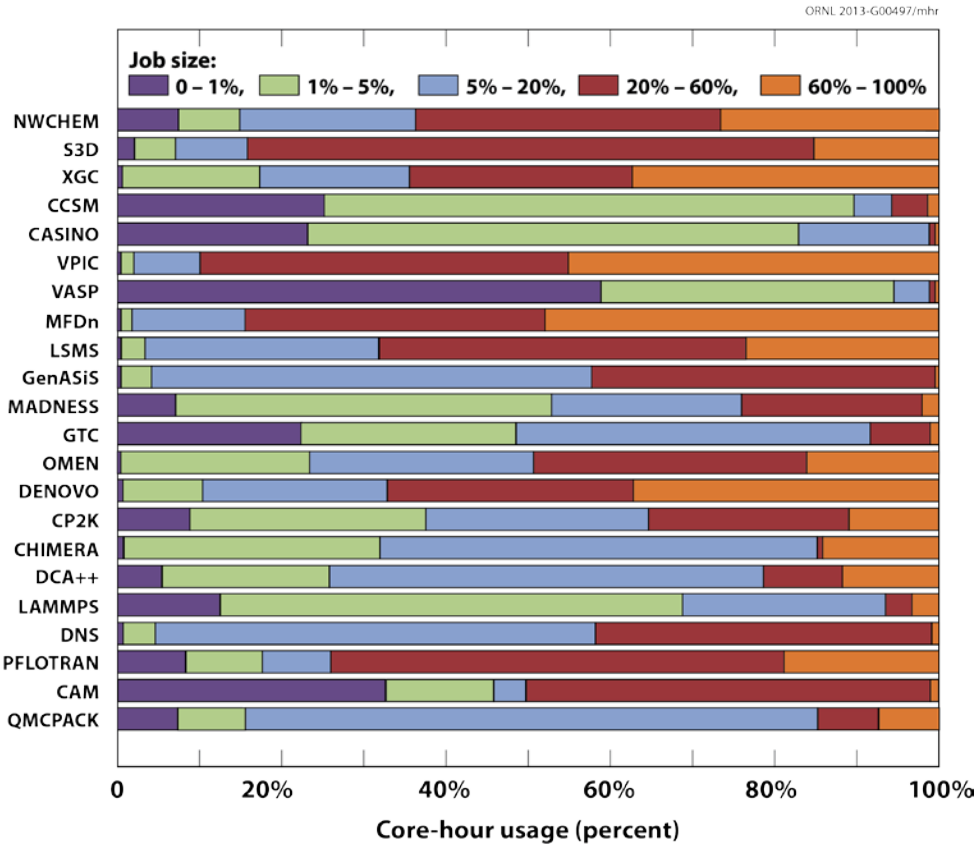


Figure 10. Scalability characteristics of selected applications.

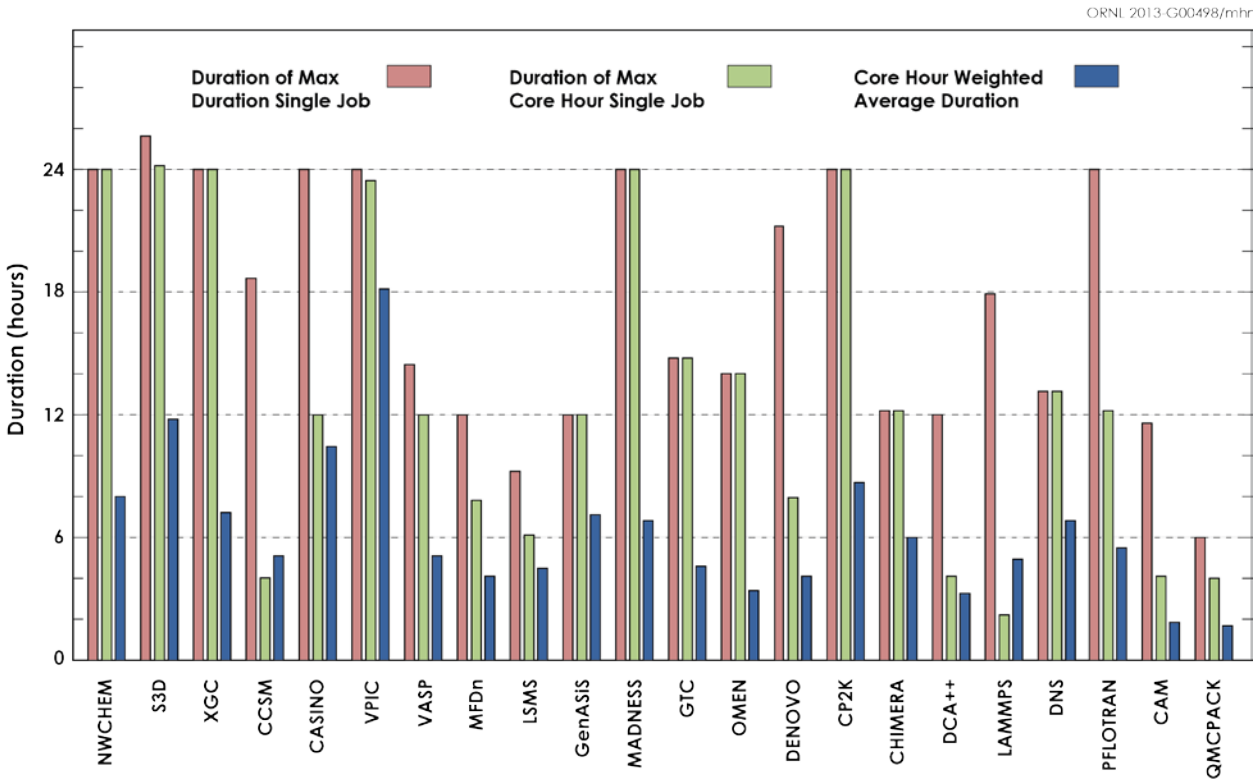


Figure 11. Job duration characteristics of selected applications.

Finally, Figure 11 shows several job duration characteristics for each of these applications, including the longest job ever run with that code, the duration of the job using the largest number of core-hours, and the weighted average job duration for the application. Many applications run on average far shorter than the scheduler maximum time limit of 24 hours. However, applications may be required to limit their runtimes even further as hardware failure rates are likely to increase for future systems, barring use of alternatives such as fault-resilient algorithms.

3.4 Summary

OLCF systems are used to process workloads from a diverse range of science domains. A large fraction of the core-hours used is spent on leadership jobs requiring at least 20% of the system to run, and these job size demands are growing. Users employ a diverse range of compilers, and library use is largely focused on a handful of heavily used libraries. The set of applications run on OLCF systems is also concentrated in a comparatively small number of highly used codes that in aggregate span the entire range of scaling regimes on OLCF systems. These applications have variegated usage of fundamental algorithm motifs and also differ greatly in their ability to scale to the entire system.

4. Science Application Requirements

4.1 Introduction

As part of the OLCF-4 requirements process, the OLCF surveyed 21 application code teams representing 18 projects across 15 science domains using leadership compute systems at the OLCF (see Appendix). This quantity is a sampling of the total roster of center projects, which includes 31 active OLCF INCITE projects representing about 60% of the resources allocated on OLCF systems, as well as ALCC and Director's Discretion projects, thus giving a cross section of center projects. Results from the survey are presented in this chapter.

Numerous workshops in recent years have determined the need for exascale computing for reaching important science discovery goals [1]. The general findings of this study show that the demand for increasing compute resources continues unabated in spite of the challenges from the disruptive shifts in computing architectures. The respondents overwhelmingly indicated that progress toward science goals could not be made at all or as fast without deployment of the next generation of computing system, which will be significantly larger than the current system of roughly 30 petaflops. This capability is required in order to perform new science and to improve the accuracy and fidelity of continuing science modeling efforts.

4.2 Hardware feature requirements

Each science code run on a leadership system involves a combination of algorithms and data sizes that have unique behavioral characteristics and impose specific requirements on system hardware, such as the need for high flop rates, large memory bandwidth, low communication latencies, or combinations of such features. OLCF projects were polled regarding their codes' needs for high performance with respect to each of these hardware characteristics for the next system.

Each team rated the importance of each hardware feature to code performance on a scale of 1 (not important) to 5 (very important). The average results from the survey are given in Table 7.

Table 7. Ranked importance of hardware characteristics

Hardware feature	Ranking	Hardware feature	Ranking
Memory bandwidth	4.4	Wan network bandwidth	3.7
Flops	4.0	Memory latency	3.5
Interconnect bandwidth	3.9	Local storage capacity	3.5
Archival storage capacity	3.8	Memory capacity	3.2
Interconnect latency	3.7	Mean time to interrupt	3.0
Disk bandwidth	3.7	Disk latency	2.9

In view of the diversity of projects represented, it is not surprising that every hardware feature shown is indicated to be of some importance for the OLCF application workload. The science domains, codes, and algorithms represented are diverse, requiring that every hardware feature give

good performance in order for science to progress, that is, that the system be architecturally well-balanced.

Several hardware characteristics stood out as being particularly important for user codes:

- The leading requirement expressed by users was memory bandwidth. Whereas many successful petascale applications in the past have had high computational intensities due to algorithm classes such as dense linear algebra, many codes are also memory intensive. Innovations such as 3D stacked memory may help such codes in the future by increasing memory bandwidth to compute cores. Also, it is unclear how much room is left for optimizing the memory access patterns of existing codes or for replacing the algorithms with less-memory-intensive algorithms. Notwithstanding, headwind is expected going forward, as the energy cost of data transfer becomes a growing concern.
- As is typical for science simulations, increasingly high floating point calculation rates are required by the codes. Unlike other application workloads such as graph algorithms, which are more communication limited, science simulations run at the OLCF require large numbers of flops to perform the needed calculations.
- Interconnect bandwidth is viewed as important. Though some codes are strong scaling codes and at large node counts are interconnect latency limited, many codes are still in a weak scaling regime and require communication of more and more data as the resolution of simulations grows; thus the amount of data per compute node to be communicated increases.
- Archival storage is a growing need. As described later in this chapter, users increasingly desire to store, analyze, and share large amounts of data related to their simulations, which is challenging with extremely large data sizes.
- Factors such as memory capacity and mean time to interrupt are not presently perceived as concerns, though downward pressures on memory size per flop and hardware failure rates may change this situation depending on the characteristics of future hardware.

4.3 Parallelism requirements

On the approach to exascale systems and beyond, one concern is the potential attrition of applications able to exploit exascale resources due to the exhaustion of available parallelism. Though some codes are beginning to face limits to scaling by conventional means, most respondents indicated that a significant amount of additional parallelism can still be extracted from their science codes. The breakdown of responses is shown in Figure 12.

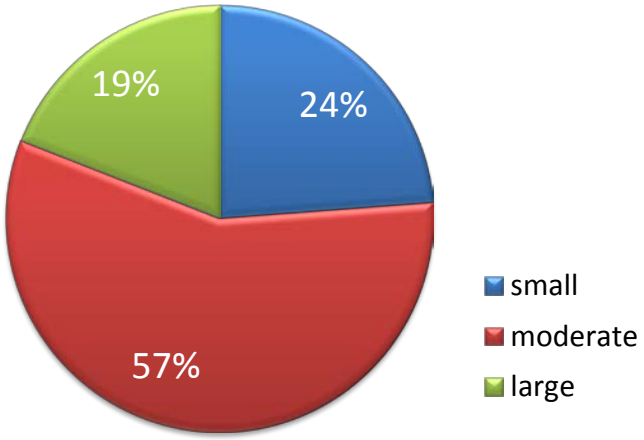


Figure 12. Estimate of available parallelism in application codes, by number of respondents.

The mechanisms that developers will use to exploit this additional parallelism run the gamut of MPI, OpenMP, CUDA, OpenACC, and libraries, and in a few cases, OpenCL or Pthreads. In general, increasing the amount of parallelism by threading, whether on CPU or GPU, is considered to be a priority.

At the same time, code teams view extracting this parallelism as a challenge. Reasons cited include the shortage of available developers, the weaknesses of application programming interfaces (APIs) for parallel programming, and the level of effort required to restructure their codes.

4.4 Programming requirements

The capabilities of science applications are ultimately limited by the labor costs for the developers who design, implement, test, and maintain the codes. Therefore, it is of paramount importance that the developer-facing programming environments and parallel programming APIs for leadership-class systems permit developers to make effective use of their time and effort developing and supporting codes.

Users were asked to rate the level of difficulty they experience in exploiting advanced heterogeneous node hardware. The results are shown in Figure 13. A total of 85% of projects rated the difficulty level as moderate to high. Some reasons given included the high level of effort required for managing their code base, limited personnel, and the disparity of the varied architectures targeted and the associated programming interfaces. Many of these concerns are common across all system vendors [40], such as the need to restructure codes for more threading and locality of memory reference, which comprises the bulk of the porting effort; mapping work to increase the numbers of cores on the node; effective vectorization; and potential fragmentation of the code base to maintain high performance across disparate platforms.

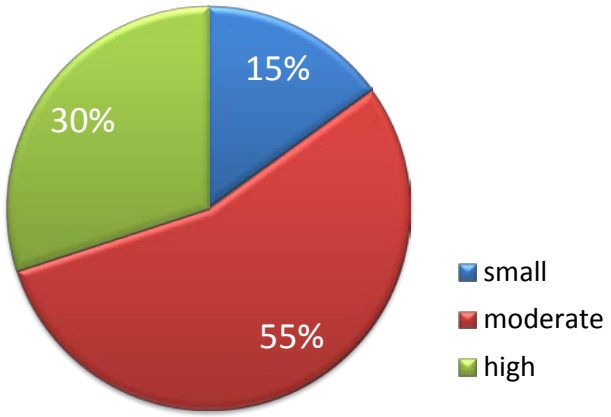


Figure 13. Assessment of difficulty in exploiting advanced hardware, by number of respondents.

In spite of this challenge, about three out of every four respondents felt their codes were moderately or very well positioned for future architectures, with moderate to high code adaptability (Figure 14). The remaining respondents indicated that significant rewriting or refactoring will be required for their codes. Nearly all respondents indicated a very strong willingness to adapt their codes to future architectures in order to get significant performance gains. However, various concerns were also mentioned, such as the lack of performance portability due to contrasting hardware and parallel APIs, the immaturity of some programming models, the lack of developer personnel, the sheer level of effort required, the number of lines of code in their code base, and the current structure of their codes.

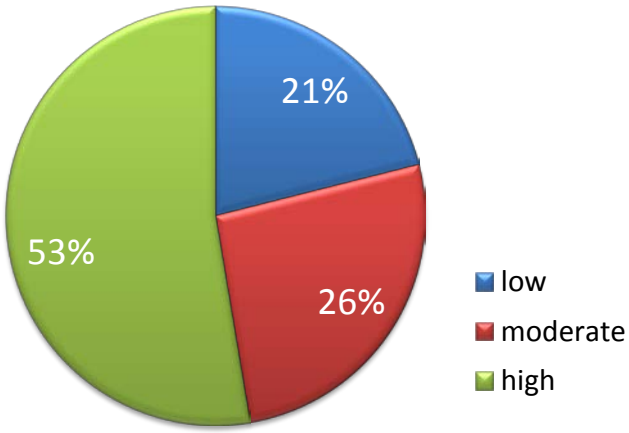


Figure 14. Assessment of code adaptability, by number of respondents.

Developers also commented on related topics:

- Of the 21 respondents, 15 indicated that they had adopted some form of CPU threading within their codes, and 16 said their codes had at least some form of accelerator usage (Figure 15). As has been observed throughout the broader community, the support for usage of accelerators to achieve performance gains has been remarkably high. Furthermore, nearly all projects indicated current support for, or interest in, supporting CPU threading.

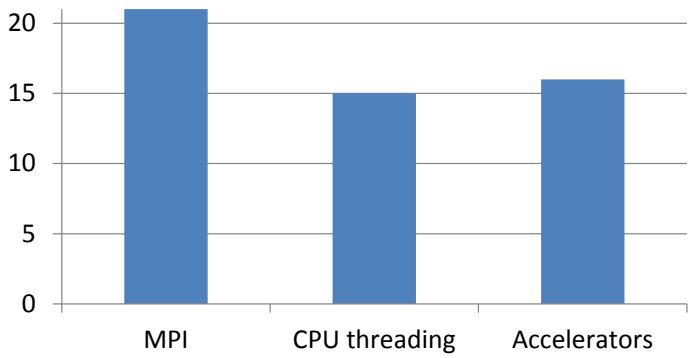


Figure 15. Current code levels of parallelism, by number of respondents.

- About half the respondents indicated that some modifications of data structures are still required to improve performance or adapt to new hardware.
- Only six respondents cited the importance of programming explicitly to the memory hierarchy with techniques such as cache blocking. This points to the importance of vendors providing means for explicitly programming to the memory hierarchy but also allowing its efficient use to be automated, as with traditional caches. However, this also highlights a potential performance gap for many projects, since the ability of the compiler and other tools to automatically restructure code for memory hierarchies is very limited, so some programmer intervention will often be needed to produce efficient code.
- Nearly all users of accelerators want to control them explicitly, for example using CUDA. The pervasiveness of GPUs and CUDA has created a low barrier to entry for programmers. Additionally, users were about equally split regarding desire to use directives-based approaches such as OpenACC, with some expressing a "wait and see" attitude.
- There is presently little interest in out-of-core techniques, improved checkpoint/restart, and explicit fault tolerance handling. No immediate pain point is driving interest in these topics, though this may change depending on the characteristics of future systems.
- Users are overwhelmingly concerned about the challenge of performance portability. The high development and maintenance effort required to tune to multiple platforms is considered a large burden, taking time and resources that might otherwise be spent on other aspects of the projects. As a result, developers may either limit the number of platforms their codes are ported to or limit how well their codes are optimized for specific platforms of interest.
- Most code teams use debugging and code optimization tools to some extent, for example, CrayPAT, Vampir, TotalView, DDT, TAU, or NVIDIA Visual Profiler. Many expressed concerns regarding missing capabilities needed for their development workflows, such as scalability of tools; usability for thread computing, GPUs, and other complex hardware; and robustness, ease of use, and readability of output.

4.5 Data Requirements

The effective storage, analysis, transfer, and sharing of data is a growing requirement for OLCF users, as the number of computations performed annually on HPC systems increases exponentially in the approach to exascale and as teams grow bigger and communities become more global.

Respondents struggled with precisely estimating the storage levels that might be required due to new physics and higher resolution simulations in the 2017 timeframe. Nonetheless, the estimates of future data storage requirements for this subset of OLCF users were substantial, going far beyond the current capabilities of the 40 petabyte OLCF Spider2 file system and 30 petabyte archival storage (Table 8). It is also important to note that bringing new big data capabilities to scientists will enable the possibility of new types of science discovery that cannot be foreseen from current requirements estimates.

Table 8. Total reported future system data requirements by users surveyed

Future data requirement	Size
Scratch	24 PB
Archival	164 PB
Runtime I/O	78 PB
Post runtime	25 PB
Avg. data lifetime	10 years

Respondents overwhelmingly expressed the importance of being able to share their data effectively. The average lifetime of archival data was estimated on average at 10 years.

The tools users currently employ for data storage, transfer, and analysis include HPSS, GridFTP, Globus Online, ESG, VisIt, ParaView, MATLAB, Ensight, Tecplot, and custom-written code. Requirements for future data analysis and management include well-parallelized and optimized analysis tools, faster transfer speeds into and out of the center, and better collaboration and repository tools.

4.6 Conclusions

Users expressed their requirements on a variety of topics in the OLCF-4 requirements survey. Users overwhelmingly affirm their need for such a system to be deployed in order to make progress toward their science goals. It is believed that many applications have significant additional parallelism that can be extracted for next-generation systems. The highest ranking hardware characteristics for OLCF-4 are increased flop rate and increased memory bandwidth, but users also require a well-balanced system that moves all hardware capabilities forward. Developers readily adopt a “do whatever it takes” approach to programming emerging new hardware architectures; however, they also express substantial concerns regarding the effort required to program disparate types of architecture efficiently. Finally, the efficient analysis and management of increasingly large amounts of data is a growing need for OLCF center users.

5. OLCF-3 Lessons Learned

5.1 Introduction

Reaching exascale will not be possible by following a "business as usual" approach to hardware design but will require substantial innovations [10]. These changes in supercomputer hardware will have significant impacts on science application development going forward. Codes will be required to support billions of compute threads. The cost of data motion will make attention to data locality increasingly important. Applications will need to navigate systems with increasing likelihood of component failures. Moreover, the power requirements of applications will need to be adjusted to fit within realistic power envelopes.

The impact of these factors is already being felt. Processor hardware advances such as NVIDIA and AMD GPUs and the Intel Xeon Phi architecture are locating increasing numbers of compute cores on a die, each with growing numbers of compute threads, thus requiring more node-level parallelism than ever before. Also, deepening cache hierarchies and the presence of off-processor accelerators are accentuating the need to pay much closer attention to data motion for designing algorithms and codes.

As a first step toward solving the challenges of exascale, ORNL deployed the OLCF-3 Titan system in late 2012 [4]. Titan is a Cray XK7 system using NVIDIA Kepler K20X GPUs with an aggregate peak speed of 27 petaflops, the first multi-petaflop system in the United States to be based on GPU technology and the top system on the November 2012 TOP500 list [5].

To effectively address DOE science goals, it was necessary not only that Titan be architecturally innovative to address emerging exascale challenges but also present a programming environment that would enable users to exploit Titan's capabilities. The OLCF has partnered with vendors to deliver development tools such as compilers, debuggers, and profilers to enable science applications to be ported effectively to the new hardware. Though a small portion of the application-porting effort is specific to the Titan platform, our experience has shown that most of the effort is applicable to other pre-exascale hardware, since the bulk of the porting work centers on the key goal of restructuring code for more parallelism and improving the locality of memory reference—issues that must be faced as a necessary step for all applications that hope to prepare for exascale.

As an exercise in application readiness, the OLCF selected pioneer applications for an early porting effort to Titan, beginning more than two years in advance of delivery of the final system. These codes were chosen to cover a diverse range of science problems and algorithm types. The application readiness process was intended not only to prepare codes for readiness to run when the machine was delivered but also to provide a body of experience in best practices that could be shared with other users of the center as well as the broader user community.

In this chapter we describe the Titan system, the applications selected for early readiness, the process of porting each of the applications, and the lessons learned in this process.

5.2 Titan system overview

Titan is an 18,688 compute node Cray XK7 system. Each node is equipped with a 16-core 2.2 GHz AMD Interlagos processor. Additionally, each node has an NVIDIA Kepler K20X GPU processor with peak performance of 1.331 TF double precision. The aggregate peak speed of the machine is approximately 27 PF. Technical specifications of the system are given in the Table 9.

Table 9. Titan system characteristics

Compute nodes	18,688
CPU	2.2 GHZ AMD Interlagos
Memory per node	32 GB DDR3-1600 / 6 GB GDDR5
Interconnect	Gemini 3D torus
GPU	NVIDIA Kepler K20X

Titan was delivered in two phases. In Phase 1 during the latter part of 2011, the ORNL Jaguar system was upgraded to XK6 compute nodes including GPU sockets that were unpopulated except for 960 nodes with NVIDIA Fermi M2090 GPUs for early application development. Then, in Phase 2 during the latter part of 2012, NVIDIA Kepler GPUs were installed on all Titan nodes.

Titan's GPUs can be utilized with several different programming approaches. First, the CUDA or OpenCL programming APIs can be used. These minor extensions to the C or Fortran programming languages allow a relatively direct level of control of instruction execution on the GPU. Alternatively, compilers supporting directives such as OpenACC [41] recognize user-supplied directives to control execution and data motion pertaining to the accelerator. The directives-based approach is potentially less invasive to user code and allows portability since the directives are embedded in comments, whereas CUDA and OpenCL allow a more "close to the metal" approach with potentially higher performance, depending on the application. A third approach suitable for some applications is the use of GPU-capable libraries.

Table 10. Titan application development software stack

Operating system	Cray CLE Linux
Parallel APIs	MPI, OpenMP, CUDA, OpenCL, GPU compiler directives, Co-Array Fortran, UPC, SHMEM
GPU-capable compilers	PGI, Cray, CAPS HMPP, NVCC, PathScale
Debugging tools	Allinea DDT
Profiling tools	VAMPIR, CrayPat, NVIDIA CUDA Visual Profiler, PGI Graphical Performance Profiler, CAPS HMPP Performance Analyzer, TAU
GPU-based libraries	CULA, Magma, FFTW, PETSc, Trilinos, BLAS 1,2,3, cuBLAS/cuSPARSE, libsci_acc
CPU libraries	HDF5, NetCDF, pNetCDF, FFTW, Cray CSML, PETSc, Trilinos, ScaLAPACK, BLAS 1,2,3, Global Arrays, libsci

To support multiple code development approaches, the Titan software environment includes the components shown in Table 10. The aim is to provide a wide variety of programming tools for heterogeneous computing hardware, to address the needs of a diverse application developer community.

5.3 Early readiness applications

Multiple criteria were used to select the science applications chosen for Titan early readiness [42]:

- **Science:** broad coverage of science domains; alignment with DOE and U.S. science missions; science results, impact, and timeliness.
- **Implementation:** wide coverage of programming models and languages, algorithms, data structures, and library requirements.
- **User community:** broad institutional, developer, and user involvement; good representation of current and anticipated DOE INCITE program workload.
- **Development processes:** mixture of easy and difficult porting challenges; availability of representative code development personnel with adequate skills and experience to engage in the activity.

Based on these criteria, the OLCF selected the following five applications for Titan early readiness:

- **CAM-SE:** a global atmospheric modeling code for weather and climate [43].
- **Denovo:** a radiation transport code for advanced nuclear reactor design [44].
- **LAMMPS:** a molecular dynamics code for modeling of materials [45].
- **S3D:** a direct numerical simulation code for modeling combustion processes [46].
- **WL-LSMS:** a nanoscience code for modeling behavior of magnetic materials [47].

Table 11. Characteristics of early readiness applications for Titan

Application	Science area	Algorithms	Data structures	Programming language	Lines of code	Libraries
CAM-SE	Climate	Spectral elements, sparse/dense linear algebra, particles	Structured grids	F90	500K	Trilinos
Denovo	Nuclear energy	3D sweep, GMRES	Structured grids	C++, F90, Python	46K	Trilinos, LAPACK, SuperLU, Metis
LAMMPS	Materials/ biology	Molecular dynamics, FFT	Particle lists	C++	140K	FFTW
S3D	Combustion	Finite differences, dense/sparse linear algebra, particles	Structured grids	F90	10K	–
WL-LSMS	Nanoscience	Density functional theory, Monte Carlo	Small dense matrices	F77, F90, C, C++	70K	LAPACK

Table 11 summarizes the characteristics of these codes. Since the set of applications is highly diverse, the cross section of applications was expected to provide a good representation of the issues to be faced in porting petascale science applications to accelerator-based systems.

5.4 Application porting strategy

The following steps were followed to bring each application from its initial state to a point of entry into the code porting process.

- A multidisciplinary code team was set up, including an OLCF application lead, a Cray engineer, an NVIDIA developer, and others such as application developers and tool or library developers. The teams worked independently but also met regularly with the other teams and technical management to discuss progress and issues. Cross-cutting support from tool and library developers was also provided.
- A testbed GPU cluster was acquired to provide a resource for early code development and testing.
- A code inventory was performed, to assess application code structure, suitability for refactoring, algorithm structure, data structures, and data movement patterns. Also assessed were typical code use cases and problem sizes. Importantly, the execution profile and the scaling behavior were also assessed, to determine whether the code had high-usage "hot spots" (where most of the runtime was spent) that might be suitable candidates for porting to the GPU.
- A parallelization approach was determined for each application. This required determining which algorithm components of the code to port to the GPU, what problem dimensions to map to the GPU thread hierarchy, and how data movement to and from the GPU and between the GPU processors and memory would be scheduled.
- A GPU-based programming model for the code port was decided, whether CUDA, OpenCL, directives, use of libraries, or a combination.
- A code development strategy was determined. Issues addressed included rewriting versus refactoring, portability to other platforms, incorporation of GPU code support into the build system, and relationship to the code repository main trunk.
- Representative test problems were formulated to guide the porting and code optimization process, and performance metrics were formulated for measuring success.

The port of each application posed a unique set of challenges. Following is an overview of the porting effort for each code.

CAM-SE

Execution structure: Explicit Runge-Kutta time-stepping over a 2D logically unstructured cubed-sphere grid with vertical levels; each time step with dynamical core calculations, tracer calculations, and other physics.

Execution profile: Highly problem-dependent; for the targeted cases, the tracer transport is most costly, then fluid dynamics, both requiring a vertical remap operation.

Parallelization strategy: Tracers are fully independent, thus can be parallelized on the GPU in a data parallel fashion; a new vertical remap algorithm was developed for more parallelism; arrays of structures were replaced with flat arrays; loops were fused to improved granularity; communication was improved and made asynchronous.

Programming approach: CUDA Fortran, later to move to OpenACC for better integration with the code repository trunk.

Denovo

Execution structure: Arnoldi eigenvalue solve with inner GMRES loop; matrix-vector product containing a 3D sweep operation.

Execution profile: Most time spent in a 3D sweep operation; second highest consumer is GMRES.

Parallelization strategy: Port 3D sweep to GPU, restructure algorithm to expose more parallelism and reduce data motion; port GMRES to the GPU via the Trilinos library.

Programming approach: CUDA

LAMMPS

Execution structure: Molecular dynamics forward stepping in time; particle motion at each step derived from force-field calculations.

Execution profile: A major portion of time spent in short-range force calculations; long-range force calculations a barrier to scalability.

Parallelization strategy: Port short-range force calculations, neighbor list calculations and parts of long-range force calculations to the GPU, with one or more threads per atom; use scalable MSM algorithm for long-range forces.

Programming approach: Portable CUDA/OpenCL interface.

S3D

Execution structure: Runge-Kutta time stepping on 3D structured grid with explicit finite differences.

Execution profile: Most execution time spent in reaction rate, right-hand-side, and transport coefficients calculations.

Parallelization strategy: The code was restructured, moving a 3D loop up the call tree to expose coarser-grained parallelism.

Programming approach: Initial port of kernels to CUDA, followed by a full port to OpenACC.

WL-LSMS

Execution structure: A master nodes spawns Monte Carlo "walkers" that are independent and require occasional synchronization.

Execution profile: Nearly all work is concentrated in small complex matrix inversion and matrix-matrix products.

Parallelization strategy: Refactor code to allow multiple atoms per MPI task, thus more threading; use library code and customized code for matrix operations on the GPU.

Programming approach: GPU library.

5.5 Performance results

Table 12 shows a set of early performance results from these applications on Titan [48]. The comparison criterion is to compare runtime using Titan nodes each containing a CPU and a GPU against a Cray XE6 with each node containing two AMD Interlagos CPUs and no GPU. The intent here is to show the value of putting a GPU rather than a CPU in the second socket of the node. As these are early results, performance is expected to improve as the codes are further tuned to the new architecture. In all cases the improvement factor compared to CPU-only is at least about a factor of two, and as high as a factor of seven. In spite of the challenge of porting these codes to a substantially different architecture, the applications in all cases have been able to take substantial advantage of the strong compute capabilities of the GPUs.

Table 12. Early Titan performance results

Application	Performance ratio, XK7 vs XE6
Cam-SE	1.8 (estimated from Fermi)
Denovo sweep	3.8
LAMMPS	7.4 (mixed precision)
S3D	1.8
WL-LSMS	3.8

l5.6 Lessons learned

Refactoring of codes for new architectures is a labor-intensive effort. Many developers are now actively migrating their codes to GPUs and other advanced hardware, and we anticipate that more code modifications will be required in coming years as disruptive hardware changes continue. The following lessons learned from the Titan readiness effort are ones that we believe will be applicable to future code migration efforts.

- The code porting work had several recurring themes: finding more threadable work for the GPU; improving memory access patterns by modifying loops and changing data structures; making GPU work (kernel calls) more coarse-grained (e.g., via loop fusion and loop permutation); making data on the GPU more persistent; porting increasing numbers of kernels to the GPU; and overlapping data transfers with other work. In this regard, it is helpful to deploy as much asynchronicity as possible by overlapping CPU work, GPU work,

MPI communication, and CPU-GPU data transfer to exploit the available hardware as much as possible.

- Codes may require optimization work prior to the port (e.g., improving MPI communications).
- Some changes may have cross-cutting impact in many files across the entire code base (e.g., data structure changes). This may be manageable by abstractions such as use of C++ templates and inlinable functions, though it is difficult to plan ahead for every possible contingency.
- Software tools were lacking at the beginning of the project, but the situation has improved. Debugging and profiling tools were useful in some cases. The lack of hardware counters on the GPU corresponding to metrics familiar to scientific computing was in some cases an impediment. Similarly, since performance is important, vendors must expose within tools, libraries, and parallel APIs the performance information to the programmer that is actionable for optimizing the code.
- The level of difficulty of the code port was in part determined by the structure of the algorithms (available parallelism, computational intensity), the code execution profile (flat vs containing performance hot spots), and the code base size (lines of code).
- A fairly uniform figure of roughly two person-years was required in each case to port the code. Though this is substantial work, the code restructuring is an essential step to prepare the codes for exascale. Furthermore, as a side benefit, the resulting codes in several cases now run approximately twice as fast as the original codes on the older CPU-only hardware due to the optimizations implemented.
- It is estimated that roughly 70% of developer time was spent in code restructuring work that was independent of the specific parallel API used, whether CUDA, OpenCL, or OpenACC. It is thus our expectation that porting to other accelerator hardware such as, for example, Intel Xeon Phi or AMD Fusion with different software APIs will now be much easier.
- All the early readiness applications are under continual active development. Porting code to the GPU can be pursuing a moving target as the science of the code may be changing.
- The tendency for flops to increase faster than memory speed for new hardware might lead us to consider new science opportunities that are enabled by this hardware at little additional cost (e.g., increasing the number of degrees of freedom per gridcell).
- For some applications it is a struggle to find enough efficiently usable parallelism. Developers may need to look in unconventional places for some codes to find enough parallelism for exascale, such as possibly parallelism in time.

5.7 Conclusions

The Titan application early readiness effort has successfully enabled the targeted applications to exploit GPU-equipped nodes to generate new science. Other community codes have also been ported to Titan and are showing effective use of GPUs [49]. The developer community has been extraordinarily willing to embrace accelerator technology on account of the performance advantages it brings. However, the programming environment for accelerators must mature,

standards must be widely embraced by vendors, and performance portability among current and future systems must be enabled for developers.

6. Conclusions

The OLCF-4 100–200 petaflop system planned for the 2017 timeframe will be a follow-on system to the OLCF's entry into heterogeneous computing via Titan and also precede the delivery of an exaflop-capable system anticipated near the end of the decade. This report documents our findings regarding application requirements for this pre-exascale system. Our primary findings follow.

- The science teams surveyed affirm the essential need for such a system, without which progress toward their science goals would be impossible or significantly impeded.
- Users of the OLCF exploit a diverse set of science applications, models, and algorithms. This necessitates a well-balanced system that is not deficient in any essential performance characteristic.
- While some codes are having difficulty scaling further, in aggregate the codes in use on OLCF systems are scaling to increasingly high core counts. Furthermore, most application teams indicate that moderate to high amounts of parallelism can still be extracted from their applications.
- Increased main memory bandwidth and higher total flop rates are the most valued hardware characteristics for generating new science on the next-generation system.
- The community has expressed great willingness to exploit new hardware, including heterogeneous architectures, due to the opportunities provided for improved performance.
- At the same time, the community has expressed substantial concerns about programmability and performance portability as relevant issues for protecting the investment they have made in their codes.
- The need to effectively store, transfer, and analyze the large quantities of data connected with science simulations is expected to grow tremendously.
- Preparing applications for next-generation systems that entail a significant architectural shift requires substantial well-planned effort, but the resulting ported applications on the new system hardware yield significant performance gains. Vendor support and access to early hardware are impactful in this regard.

The proliferation of new system architectures and programming models that is fueling the upward growth of HPC capabilities is posing significant challenges to the science applications. Future systems must provide the means necessary to move these applications forward to exascale and beyond.

Acknowledgments

The authors thank ORNL staff members who provided material for this report, including Ashley Barker, Buddy Bland, Chris Fuson, Scott Klasky, Bronson Messer Bill Renaud, and Shiquan Su. The authors also thank Hai Ah Nam for her assistance as reader and Amy Harkey for her assistance as editor.

This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC0500OR22725. This document describes research performed under contract DE-AC0500OR22750 between the U.S. Department of Energy and Oak Ridge Associated Universities.

References

[1] *Exascale Workshop Panel Meeting Report,* report from meeting held Jan. 19–20, 2010, Washington, D.C., U.S. Department of Energy Office of Advanced Scientific Computing Research, http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Trivel_piece_exascale_workshop.pdf.

[2] *DOE Leadership Computing Facility's Strategic Plan for Advancing Fundamental Discovery and Understanding in Science and Engineering over the Next Decade, 2014–2023,* Oak Ridge National Laboratory and Argonne National Laboratory, March 1, 2013.

[3] Wayne Joubert, Douglas Kothe, and Hai Ah Nam, *Preparing for Exascale: ORNL Leadership Computing Facility Application Requirements and Strategy*, ORNL/TM-2009/308, Oak Ridge National Laboratory, December 2009, https://www.olcf.ornl.gov/wp-content/uploads/2010/03/olcf-requirements.pdf.

[4] Oak Ridge Leadership Computing Facility. Introducing Titan: Advancing the Era of Accelerated Computing, http://www.olcf.ornl.gov/titan/.

[5] Top500 project, http://top500.org.

[6] W. J. Harrod, "Thinking about the future of large-scale computing," presented at SC12, Salt Lake City, Utah, Nov. 10–16, 2012, http://sc12.supercomputing.org/schedule/event_detail.php?evid=inspkr104.

[7] "'No exascale for you!' an interview with Berkeley Lab's Horst Simon," *HPCwire,* May 15, 2013, http://www.hpcwire.com/hpcwire/2013-05-15/no_exascale_for_you_an_interview_with_nersc_s_horst_simon.html?featured=top.

[8] U.S. Department of Energy Office of Science, "Scientific Discovery through Advanced Computing (SciDAC)," last modified Mar. 15, 2013, http://science.energy.gov/ascr/research/scidac/co-design/.

[9] Lawrence Livermore National Laboratory, "DOE extreme-scale technology acceleration: FastForward," last modified May 17, 2013, https://asc.llnl.gov/fastforward/.

[10] *ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems,* Peter Kogge, ed., Defense Advanced Research Projects Agency, 2008, http://users.ece.gatech.edu /~mrichard/ExascaleComputingStudyReports/ECS_reports.htm.

[11] F. J. Seinstra et al., "Jungle computing: distributed supercomputing beyond clusters, grids, and clouds," Chapter 8, pp. 167–197, in M. Cafaro and G. Aloisio, eds., *Grids, Clouds and Virtualization*, Springer-Verlag London, 2011, www.cs.vu.nl/~fjseins/Papers/Other /gcv-book-chapter.pdf.

[12] Herb Sutter, "Welcome to the jungle," http://herbsutter.com/welcome-to-the-jungle/.

[13] *Inter-Agency Workshop on HPC Resilience at Extreme Scale,* Feb, 21–24, 2012, National Security Agency Advanced Computing Systems, http://institute.lanl.gov/resilience/docs /Inter-AgencyResilienceReport.pdf.

[14] D. Heaton, J. Carver, R. Bartlett, K. Oakes, and L. Hochstein, "The relationship between development problems and use of software engineering practices in computational science & engineering: a survey," *Proc. 1st Int. Workshop on Maintainable Software Practices in e-Science,* held during eScience 2012, Chicago, IL, http://www.software.ac.uk/sites/default/files /softwarepractice2012_submission_10.pdf.

[15] Jeremy Geelan, "Moore's law: 'we see no end in sight,' says Intel's Pat Gelsinger," *SOA World Magazine*, May 1, 2008, http://java.sys-con.com/node/557154.

[16] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Proc. 38th Int. Symp. on Computer Architecture (ISCA '11),* ACM Press, 2011, ftp://ftp.cs.utexas.edu/pub/dburger/papers/ISCA11.pdf.

[17] Vivek Sarkar, "ASCAC Subcommittee on Synergistic Challenges in Data-Intensive Science and Exascale Computing," ASCAC Meeting, Washington, DC, Mar. 5, 2013, http://science.energy.gov /~/media/ascr/ascac/pdf/meetings/20130305/Data-subcommittee-ASCAC-presentation-March-2013-v2.pdf.

[18] Federal Plan for High-End Computing: Report of the High-End Computing Revitalization Task Force (HECRTF), May 10, 2004, http://www.csci.psu.edu/docs/hecrtf.pdf.

[19] U.S. Department of Energy Strategic Plan, DOE/CF-0067, May 2011, http://energy.gov/downloads/2011-strategic-plan.

[20] DOE Strategic Plan 2012 GPRA Addendum, http://energy.gov/sites/prod/files/ DOE%20Strategic%20Plan_2012%20GPRA%20Addendum.PDF.

[21] DOE Office of Science Strategic Plan, Feb. 2004, http://stratml.hyperbase.com/DOEER/DOEER.html.

[22] Advanced Scientific Computing Research: Delivering Computing for the Frontiers of Science: Facilities Division Strategic Plan for High Performance Computing Resources, 2007,

http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Ascr_facilities_strategic_plan.pdf

[23] U.S. Department of Energy, Office of Science, High Performance Computing Facility Operational Assessment, CY2011 Oak Ridge Leadership Computing Facility, Feb. 2012, http://info.ornl.gov/sites/publications/files/Pub35226.pdf.

[24] ASCAC Subcommittee Report: The Opportunities and Challenges of Exascale Computing, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[25] Discovery in Basic Energy Sciences: The Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[26] Forefront Questions in Nuclear Science and the Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[27] Fusion Energy Sciences and the Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[28] Opportunities in Biology at the Extreme Scale of Computing, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[29] Challenges for the Understanding the Quantum Universe and the Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[30] Challenges in Climate Change Science and the Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[31] Science Based Nuclear Energy Systems Enabled by Advanced Modeling and Simulation at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[32] Scientific Grand Challenges in National Security: The Role of Computing at the Extreme Scale, http://science.energy.gov/ascr/news-and-resources/program-documents/.

[33] Exascale Workshop Meeting Report January 19-20, 2010, http://www.exascale.org/mediawiki/images/4/48/TrivelpieceExascaleWorkshop.pdf.

[34] A Workshop to Identify Research Needs and Impacts in Predictive Simulation for Internal Combustion Engines (PreSICE), March, 2011, http://www1.eere.energy.gov/vehiclesandfuels/pdfs/presice_rpt.pdf.

[35] Basic Research Needs for Clean and Efficient Combustion of 21st Century Transportation Fuels, DOE Office of Science, 2006, http://science.energy.gov/~/media/bes/pdf/reports/files/ctf_rpt.pdf.

[36] DOE INCITE Leadership Computing, "Guide to HPC," http://www.doeleadershipcomputing.org/guide-to-hpc/.

[37] Mark Fahey, Nicholas Jones, and Bilel Hadri, "The Automatic Library Tracking Database," *Proc. CUG 2010,* May 2010, https://cug.org/5-publications/proceedings_attendee_lists/CUG10CD/pages/1-program/final_program/CUG10_Proceedings/pages/authors/01-5Monday/4A-Fahey-paper-CUG%202010%20-%20ALTD.pdf.

[38] Wayne Joubert and Shiquan Su, "An analysis of computational workloads for the ORNL Jaguar system," *ICS'12,* June 25–29, 2012, San Servolo Island, Venice, Italy.

[39] P. Colella, "Defining software requirements for scientific computing," DARPA HPCS presentation, 2004.

[40] J. Levesque, "Hybrid multi-core programming for exascale computing," CPS 2011, July 27, 2011, http://www.olcf.ornl.gov/wp-content/uploads/2011/08/TitanSummit2011 _Levesque.pdf

[41] OpenACC Application Program Interface, http://www.openacc-standard.org/.

[42] Bronson Messer, "Early science at the OLCF on Titan," Advanced Scientific Computing Advisory Committee (ASCAC) Meeting, Washington, D.C., Aug. 14–15, 2012, http://science.energy.gov /~/media/ascr/ascac/pdf/meetings/aug12/ASCAC-8-2012-Bronson.pdf.

[43] M. Norman, L. Larkin, R. Archibald, I. Carpenter, V. Anantharaj, and P. Micikevicius, "Porting the community atmosphere model spectral element code to utilize GPU accelerators," *Proc. of Int. Cray User Group Meeting,* Hamburg, 2012.

[44] Christopher G. Baker, Gregory G. Davidson, Thomas M. Evans, Steven P. Hamilton, Joshua J. Jarrell, and Wayne Joubert, "High performance radiation transport simulations: preparing for Titan," *Proc. SC12,* Salt Lake City, Utah, Nov. 10–16, 2012.

[45] W. M. Brown, "Porting LAMMPS to the Titan system," presented at Cray Technical Workshop on XK6 Programming, Oak Ridge, TN, Oct. 9–10, 2012, www.olcf.ornl.gov/wp-content/uploads /2012/05/brown_cray_tech_12.pdf (see https://www.olcf.ornl.gov/training-event /cray-technical-workshop-on-xk6-programming/).

[46] J. Levesque, "Using the Cray programming environment to convert an all MPI code to a hybrid-multi-core ready application," presented at Cray Technical Workshop on XK6 Programming, Oak Ridge, TN, Oct. 9–10, 2012, http://www.olcf.ornl.gov/wp-content/training /CrayTech_XK6_2012/CTW_S3D_10_10.pdf (see https://www.olcf.ornl.gov/training-event/cray-technical-workshop-on-xk6-programming/).

[47] D. Nicholson and M. Eisenbach, "Preparing WL-LSMS LSMS for first principles thermodynamics calculations on accelerator and multicore architectures," presented at Cray Technical Workshop on XK6 Programming, Oak Ridge, TN, Oct. 9–10, 2012, http://www.olcf.ornl.gov /wp-content/uploads/2012/05/WL-LSMS-GPU_don1.pdf (see https://www.olcf.ornl.gov /training-event/cray-technical-workshop-on-xk6-programming/).

[48] Arthur S. Bland, "Early experience with the Titan system at Oak Ridge National Laboratory," presented at SC12, Salt Lake City, Utah, Nov. 10–16, 2012, http://sc12.supercomputing.org /schedule/event_detail.php?evid=mswk111, slides at http://developer.download.nvidia .com/GTC/PDF/GTC2012/PresentationPDF/BuddyBland_Titan_SC12.pdf.

[49] Bronson Messer, "Application readiness at ORNL," presented at Cray Technical Workshop, Oct. 9, 2012, http://www.olcf.ornl.gov/wp-content/uploads/2012/05/CrayTechOct2012 Bronson2.pdf.

APPENDIX.
APPLICATION REQUIREMENTS SURVEY

Members of the Scientific Computing Group at the Oak Ridge Leadership Computing Facility (OLCF), part of the National Center for Computational Sciences (NCCS) at Oak Ridge National Laboratory (ORNL), surveyed numerous scientists in a broad range of scientific domains and asked them to speculate on requirements for their scientific application(s) on Leadership Computing platforms in the next 3–5 years. A large fraction of the information, guidance, and plans outlined in this document is derived from the answers provided in these surveys from this expert community of leading computational scientists. The survey questions are listed below.

Science Drivers

- Why does your science need leadership computing in 2016?
- Without leadership computing, can progress be made at all? Or as fast?
- What science questions will you be answering in 2016?
- What impact will your answers have on your field? Other fields?
- How will you use your results to confirm observations or measurements (e.g., are you simulating a particular experimental device, or will your findings be tested in other ways)?

Appropriateness of Current Code Base for Future Architectures

- Do you envision writing a completely new code over the next three years in order to accomplish your scientific and/or technical goals (even if that future code uses components from your current code base)? Or do you consider that your current code base is well positioned to achieve your technical goals through evolutionary and/or incremental changes over the next three years?

Hardware Features

For the following computing hardware characteristics, please rank on a scale of 1 to 5 the importance of improving this hardware feature in order to execute your scientific and/or technical vision and goals for your application code in the 2016–2017 timeframe (5 = very important, 1 = not important). Please add additional comments to each of these items, as appropriate, below.

- flops (floating point operations per second) to perform calculations:
- memory capacity (more grid cells, particles, degrees of freedom., different algorithm, etc.):
- memory bandwidth (sparse linear algebra, limited computations per accessed data element, etc.):
- memory latency (unpredictable memory access patterns, unstructured grids, graph algorithms, etc.):
- interconnect bandwidth – the need to communicate large amounts of data between compute nodes:
- interconnect latency – the need to communicate large numbers of messages between compute nodes:
- local storage capacity – increasing disk space for saved results and restart files:

- disk bandwidth – growing impact of data storage to disk on runtime:
- disk latency – large number of writes to disk of small size:
- mean time to interrupt – need for longer compute times between restart dumps:
- archival storage capacity – long-term storage of results:
- WAN network bandwidth – need to communicate large amounts of data to/from offsite location:

Parallelism

- How much more parallelism do you believe can be extracted from your science problem, whether at the node level or thread level (e.g., using vector instructions)?
- How difficult will it be to extract this parallelism (e.g., will it require extensive changes to your data structures)?
- Do you plan to exploit additional parallelism via code you write yourself, or do you plan to use libraries to exploit this parallelism?
- Which programming models (OpenMP, OpenACC, etc.) or which libraries (MAGMA, cuFFT, etc.) do you plan to use?

Heterogeneity

- What is your application's level of adoption of heterogeneous node computing?
- What is the level of difficulty in exploiting heterogeneous hardware for your codes?

Serial Code Execution

- How much of your code is intrinsically serial, and will require fast, serial execution?

Programming Model

- How adaptable is your code to new programming models, in terms of lines of code and software design?
- Do you plan to make changes or use techniques like the following in your code? If not, why not? If so, how hard will these modifications be to complete?
  - Modifications of fundamental data structures (e.g., structure of arrays vs array of structures)?
  - Explicit memory hierarchy control (e.g., cache blocking)?
  - Fine-grained programming of CPU's through directives-based models (e.g., OpenMP or Pthreads constructs)?
  - Programming of accelerators through directives-based models (e.g., OpenACC)?
  - Explicit accelerator task/thread management via CUDA or OpenCL?
  - Explicit use of out-of-core techniques via utilization of disk or NVRAM?
  - More efficient checkpoint-restart and/or resiliency algorithms via utilization of NVRAM?
  - Explicit fault tolerance handling (e.g., via FT-MPI to detect and correct failures)?
- Do you feel that performance portability to multiple hardware platforms is a challenge?
- How do you address this currently?
- Do you use program development, optimization, and/or debugging tools today?

- What new capabilities do you anticipate needing for programming tools for using compute capabilities 10 times greater than available today?

Data Reduction, Processing, and Storage

- Please estimate your data storage requirements necessary to achieve your scientific goals in 2016–2017, broken down by scratch space and short- and long-term archival space.
- Please describe the capabilities and/or tools needed for in-situ reduction and/or analysis of computed data sets that maybe too large or too time consuming to write to disk.

Data Sharing

- How do you intend to share the scientific data emerging from your work in the 2016–2017 future?
- Will you share your data with your scientific community?
- Give an estimate of the useful lifetime of your scientific data:
- What types of tools for data storage, movement, and analysis do you currently use?
- Where do you see the need for tools development?

Runtime I/O

- What will be the total data set size for runtime checkpoint and restart?

Post-Runtime Data Analysis

- What will be the total data set size for post-runtime I/O?

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY