Planning for Pre-Exascale Platform Environment (Fiscal Year 2015 Level 2 Milestone 5216)

LLNL-TR-677460

September 28, 2015

Disclaimer

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

Table of Contents

Introduction and Background

This *Plan for ASC Pre-Exascale Platform Environments* document constitutes the deliverable for the fiscal year 2015 (FY15) Advanced Simulation and Computing (ASC) Program Level 2 milestone *Planning for Pre-Exascale Platform Environment*. It acknowledges and quantifies challenges and recognized gaps for moving the ASC Program towards effective use of exascale platforms and recommends strategies to address these gaps. This document also presents an update to the concerns, strategies, and plans presented in the FY08 predecessor document that dealt with the upcoming (at the time) petascale high performance computing (HPC) platforms.

With the looming push towards exascale systems, a review of the earlier document was appropriate in light of the myriad architectural choices currently under consideration. The ASC Program believes the platforms to be fielded in the 2020s will be fundamentally different systems that stress ASC's ability to modify codes to take full advantage of new or unique features. In addition, the scale of components will increase the difficulty of maintaining an error-free system, thus driving new approaches to resilience and error detection/correction. The code revamps of the past, from serial- to vector-centric code to distributed memory to threaded implementations, will be revisited as codes adapt to a new message passing interface (MPI) plus "*x*" or more advanced and dynamic programming models based on architectural specifics.

Development efforts are already underway in some cases, and more difficult or uncertain aspects of the new architectures will require research and analysis that may inform future directions for program choices. In addition, the potential diversity of system architectures may require parallel if not duplicative efforts to analyze and modify environments, codes, subsystems, libraries, debugging tools, and performance analysis techniques as well as exploring new monitoring methodologies. It is difficult if not impossible to selectively eliminate some of these activities until more information is available through simulations of potential architectures, analysis of systems designs, and informed study of commodity technologies that will be the constituent parts of future platforms.

Drivers and Requirements

It is critical that the ASC Program carefully identify infrastructure components most needed by customers and that efforts be well-managed as an integrated enterprise that meets users' needs. Efforts in the Computational Systems and Software Environment (CSSE) and Facility Operations and User Support (FOUS) ASC Program subprogram elements (together with the other major ASC Program subprograms (Integrated Codes (IC), Physics and Engineering Models (PEM), Verification and Validation (V&V), and Advanced Technology and Development Mitigation (ATDM)) are resources and mechanisms applied to improve stockpile understanding and decisions. ASC Program products (and the efforts that lead to them) must be structured such that they are clearly explained in a compelling way to stakeholders. In developing new tools and capabilities, output of one activity is vital to the success of others, and it is important to ensure that customer requirements are identified, analyzed, and validated.

The ASC Program must continue to engage in partnerships with other federal agencies, academia (through the Predictive Science Academic Alliance Program (PSAAP), and platform architecture vendors to develop strategies for productivity gains within realistic future budgets. The ASC Program will need to build on its “weapons science” computational environments to deliver balanced computing resources over the next decade. The Department of Energy (DOE) FastForward and DesignForward programs involve tri-lab CSSE staff through collaboration and oversight and focus on supporting environments for both existing integrated codes and new advanced codes being developed within the ATDM subprogram. Current integrated codes are validated and trusted and will remain a key tool to support the laboratory mission; the new code efforts are promising but high-risk and the road to adoption will not be immediate. To address these challenges, multiple collaborative efforts are being leveraged through co-design.[1]

Additionally, individual laboratories use non-recurring engineering (NRE) investments tied to advanced technology (AT) system procurements to impact near-term research that directly supports these advanced systems. These works will establish the technological foundation to build toward exascale computing environments, which predictive capability will demand.

Scope of this Document

This document identifies, assesses, and specifies development and deployment approaches for critical infrastructure components. It identifies and quantifies potential technical gaps or issues, and, where they exist, defines a prioritized approach to closing those gaps. While this document represents a specific FY15 Level 2 milestone deliverable, the pre-exascale infrastructure components that it describes will be deployed over the next decade, and this plan is applicable to multiple ASC pre-exascale platforms deployed during that time, including AT systems—Sierra, the AT system due for arrival in 2017, and Crossroads, due for arrival in 2020. The ASC Program requires well-integrated infrastructure components to leverage the investments that will be made in pre-exascale computer systems and to maximize their impact across the tri-lab user community. Some of these CSSE/FOUS infrastructure components will be applicable to both ASC commodity technology (CT) systems and AT systems. However, there may be situations where the user base or the problem-solving capabilities are so unique (or site-specific) that integrating specialized components into a more general-purpose simulation environment may require additional trade-off cost decisions.

The document spans five key CSSE/FOUS technical areas:

- Programming environments and tools
- System management and monitoring
- Data analysis and visualization
- Input/output (I/O), file systems, and storage
- Networks and interconnects

[1] *ASC Co-Design Strategy Document*, draft, anticipated publication date September 2015.

Each of the five technical sections provides a brief overview followed by a description of major areas of concern. Each concern is addressed by a strategy and characterized by priority and difficulty. Suggestions for partnerships within and outside of the tri-lab community for collaborative work to address these strategies are listed.

Note that the System Management and Monitoring section was included in recognition of the growing dependence on continuous system awareness and potential interactions with facilities. This relationship will only intensify with larger, more complex platforms, and operations personnel will be required to intervene more directly with the workload, scheduling, and maintenance of major systems to minimize impact on the ecosystem surrounding the computers.

Three priorities will be defined for each concern listed:

- Essential: indispensable for an exascale system to function at the most basic level
- Highly Important: necessary for system scalability reasons or required user productivity
- Important: considered highly desirable for applications or users to achieve full system performance or scalability

Three levels of difficulty will be defined:

- Hard: multiple person-years of effort with high potential for unforeseen technical challenges and no guarantee for success
- Medium: multiple person-years of effort but technically straightforward
- Easy: Technically straightforward but may require multisite or multi-agency agreement and coordination

System and Infrastructure Considerations

A well-balanced pre-exascale environment requires consideration of not only the actual hardware and software but of concepts such as reliability, availability, and serviceability (RAS), and system management and monitoring. Areas of consideration are presented below.

Processors

Over the timeframe of interest for exascale systems, Moore's law is predicted to come to an end. Moore's law encapsulates the empirical observation that the number of transistors on a die at constant cost doubles about every twenty-four months. However, Moore's law is silent on transistor performance. In the past, shrinking transistor feature sizes permitted a drop in circuit voltage and enabled an increase in the core's (central processing units (CPUs)) frequency at a rate that led to application performance doubling about every eighteen months. However, complementary metal-oxide semiconductor (CMOS) feature sizes are now so small that continued frequency increases seriously increase power consumption. On the other hand, using smaller feature size to add more cores has little effect on power consumption. The majority of current and expected future microprocessor performance will come from a geometric increase in cores and not from core frequency increases. Current commodity server processors have many tens of cores, specialty processors such as the Intel Phi will have near 100 cores, and graphic processing unit (GPU) processors such as NVidia Tesla have near 3000 lightweight cores. This tread seems to only increase as exascale nears. Dealing with the high level of parallelism is a chief challenge to the ASC Program and the computing environment.

Memory

Microprocessors with 32–128 cores (CPUs) will force microprocessor designers and system architects to address balance factors for processor access to local memory hierarchies. With this many cores per processor die, it will be easy for a large fraction of the total computational capability of the processor complex to lie idle while waiting for data, and it again will force an effort to address the memory wall. Understanding balance between the need for larger and more capable memories and increased computational speed will be imperative for ASC codes to perform optimally. As the ASC Program increases physical fidelity and detail in its codes, improves numerical algorithms, and increases resolution, the need for more memory as well as more capable memory will continue to grow. Developers will need to understand what demands ASC applications make on memory subsystems and what the balance between processing and memory should be. Additional complexity is being introduced to the memory hierarchy by new memory technologies addressing the need to reduce overall power of the system. Non-Volatile Memory (NVM) and other technologies will be augmenting and replacing dynamic random-access memory (DRAM) in upcoming systems. This changes both the capacity and performance, in both bandwidth and latency, of application memory. Abstracting the complexity and simplifying the programming interface for the applications is of chief concern.

Interconnects

Scalable system architectures will also drive requirements for improvements in the interconnect fabric to take advantage of potential memory subsystem improvements. Areas for improvement include interconnect bandwidth, latency and message injection rate. In addition, this time frame may also see the development and use of optical technologies for interconnect fabrics. Future scalable system interconnect bandwidth requirements will accelerate the practical viability (price) of optics for interconnect technologies. Additional drivers include the weight of copper cables and the imposition of shorter distance limits as signaling rates increase on copper cables. More complex interconnect topologies are becoming more common, such as dragonfly, N-dimensional tori, and hierarchical topologies which will require tools and experience to understand the application related performance trade-offs.

Accelerators

Accelerators are now more closely integrated into mainstream processors, and they will have a significant impact on the ability to achieve exascale levels of performance before 2022. Incorporation of single instruction, multiple data (SIMD) or vector accelerators into scalable system compute nodes will only deliver performance if interdisciplinary efforts to develop a new generation of parallel algorithms and their associated advanced solvers can be vectorized and expose enough parallelism. The AT systems, Sierra and Crossroads, will need to leverage these technologies, and this will require advanced tools and compilers to expose the performance.

Operating Systems

Scalable system software is a critical enabling technology for future systems. The ASC Program expects to have full-service operating system (OS) software such as Linux as well as lightweight kernel (LWK) OS software. ASC invested in two systems that use LWK system software, Red Storm/XT3 and BlueGene/L, and both demonstrated scalable and reliable performance up to full scale. While pursuit of application performance at exascale and beyond may also continue to require the use of an LWK, there are users that want to use some functionality provided with a heavyweight operating system. There are vendor research efforts through FastForward and DesignForward and additionally novel OS and runtime research supported through DOE to address a broad range of approaches to span the gap between full-featured Linux OS and stripped down LWK OS and runtime system (RTS) software. The move to a large number of cores may also require development of new ways to distribute workload.

File Systems

Scalable parallel file system technologies are critical enablers for exascale systems and also as an integrating element within a simulation environment of CT and AT systems, data analysis engines, and archival systems. Looking forward, it appears the only way to achieve I/O performance targets for exascale systems is through tiered storage solutions such as burst buffers and data reduction methods such as in-situ and in-transit analysis. Additionally, object-based interfaces, or object stores, may be an attractive option in managing scientific data for provenance and reproducibility in the near future. With the consolidation of memory hierarchies, storage hierarchies, file system and archive, a simplification of this interface is needed.

Reliability, Availability, Serviceability

RAS will need improvements in capability and functionality to support the ability to run billions of threads on a single large problem. The ASC Program needs the ability for integration and communication among the OS, RTS, application software, and parallel file system when failures occur, particularly at extreme scales. As component part count for the systems drastically increase for exascale, the overall system environment, including facility services, will have to be highly resilient to failure of components.

System Management and Monitoring

The process of failure detection, identification, and correction is time consuming. Future systems will need to monitor themselves, automatically identify typical failures, and initiate corrective action. This kind of “autonomic” behavior is essential to operate more hardware without hiring more staff. Work is starting on initial steps to integrate data feeds into common monitoring frameworks to speed problem detection and identification by making a relevant dataset available to key people. With the extreme scale, this problem is becoming similar to commercial big data analytic problems, and tools in this area will be leveraged to aid in a scalable solution.

Programming Environment and Tools

Introduction and Background

Over the next five years (2015–2020), the systems to be deployed at the National Nuclear Security Administration (NNSA) laboratories (Trinity, Sierra, and Crossroads) will require major modifications in production nuclear weapons applications and the underlying software stack. The significant changes in these ATS-1 and ATS-2 architectures will necessitate a transition from the "MPI-everywhere" programming model, which ASC has relied on for the last twenty years, to newer programming models and environments. No single programming environment will be sufficient for the diverse set of ASC applications. For example, an application developed with one programming model may want to use a library developed with a different model or a solver developed by a university or commercial partner. Addressing these changes and complex environments are even more critical for ATS-3 and beyond systems, leading toward exascale computing.

There have been several DOE workshops and other studies that have identified four key exascale/pre-exascale challenges: dramatically improving power efficiency, improving resilience in the presence of increasing faults, enabling efficient data movement across deepening memory hierarchies and new storage technologies, and managing dramatically increased parallelism, especially at the node level. The concerns and connected strategic recommendations in this section are targeted to address and mitigate (where possible) the impact of all four challenges. The concerns and strategies are intended to help guide work within the ASC Program to help applications teams overcome these challenges and to identify current gaps requiring more work.

For example, power-aware and fault-aware approaches that can be considered at the application level, enabled by research and development in the application programming interfaces (APIs) to expose them, are emerging. Although the tri-lab application developers and analysts have successfully met the challenges of running applications at hundreds of thousands of threads/MPI tasks on Cielo and Sequoia, there is a clear recognition from the tri-lab user community of the new challenges in the near term with Trinity KNL-based nodes and the Sierra Power + NVIDIA heterogeneous nodes, both requiring new approaches to move data across deep memory hierarchies and manage multiple levels of parallelism.

Software solutions that address these challenges must also improve programmability, expanding the community of computational scientists who can use leadership-class platforms, and generally lower the barriers to application development and deployment. To address these challenges, DOE has initiated the ATDM subprogram with the implicit understanding that the entire code base may have to change and evolve with capability demonstrations by the time of Crossroads deployment. The ATDM subprogram envisions an evolution of not only the application programming models but also the entire workflow from the model generation to meshing to analysis to visualization to uncertainty quantification. If the application codes and workflows, along with the supporting platform software infrastructure, do not evolve to run efficiently on the

AT system architectures (for Trinity, Sierra, and Crossroads), the demand for cycles on commodity clusters running tested and proven MPI-based applications will dramatically rise, and moreover, the program will not have met the essential need of addressing the new ATS-x architectures. Therefore, the program must develop new techniques, novel designs, and advanced software architectures for the next-generation (exascale) software infrastructure.

A key requirement toward exascale is system integration across software stacks, resources, and hardware with communication and feedback loops. Although noted before by tool developers, it is becoming an important concern as the ASC Program moves towards next-generation systems, and it is due to the increasing complexity in both hardware and software stacks. Consequently, the necessary infrastructure to collect relevant information at potentially high data frequencies while maintaining efficiency at scale imposes substantial development efforts on this infrastructure. This will require integration of external sensors and missing interfaces in the hardware/software stack. Due to these requirements, there is overlap in some of the Concerns/Strategies between the Programming Environment and Tools technical area and the System Management and Monitoring technical area. Best practices would be to develop a common scalable data gathering and aggregation infrastructure to support both technical areas. It is infeasible to develop such infrastructures as one-off techniques separately for each area.

Balancing Evolutionary and Revolutionary Programming Models

The industry is moving from bulk synchronous programming models (with fixed per iterations barriers) to more asynchronous approaches. In particular, task-based/data-flow (also known as asynchronous many task (AMT)) approaches, where the scheduling of processing data and its dependencies are at the forefront, have generated interest across a wide spectrum of the community. AMT provides the benefits of a balanced data processing and throughput—as calculations are occurring and as needed and when the data is available—effectively maximizing overlap of communication and computation via over-decomposition. Models that can generate and use a dependency graph—Directed Acyclic Graph (DAG)—are emerging, as are runtimes that can execute them.

The same trend has also increased the level and variation of architectural solutions vying to provide such capabilities. Programming systems are either evolving from existing systems or being developed from scratch to offer such capabilities efficiently both on current and future platforms. However, one of the most widely used approaches is still MPI + X (X being a thread or task-based model addition), although the limitations of this approach are already showing. At the same time, existing programming models, including MPI and OpenMP, are evolving to address these new architectural challenges.

How programming models interact with nodes is also in a state of flux. Architecture differences (Xeon Phi/GPU), threads, and deeper memory hierarchies, for example, make for a complex application development and execution environment. Current and near-term models are based on an MPI + X approach, where X is typically OpenMP, OpenACC, or OpenCL. Increased vectorization capability is also continuing to evolve in the hardware. With the current drive toward many-core architectures, thread scaling and increased vector capability are of increasing importance.

CONCERN: Need for Targeted Evolution of Existing Models

Existing programming model systems, including the widely used standard models MPI and OpenMP, are not stagnant but rather, continue to evolve. The same is true for language standards, such as C++ and Fortran, which also increasingly include strategies for parallelism. It is in the interest of the ASC Program, which relies on these standards, to help ensure that future developments are targeted to meet the requirements posed by ASC code teams.

Strategy	Maintain active involvement in standardization bodies.
Partners	Standardization bodies, wider HPC community
Priority	Highly Important
Difficulty	Medium
The ASC laboratories should continue and coordinate their strong presence in the respective standardization bodies, both for parallel programming models and the respective base languages. A coordinated effort will help shape the next generation of these standards and enable a larger amount of code reuse from existing investments that already rely on these models.	

Strategy	Research new abstractions for the existing models.
Partners	LDRD, DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Hard
Where existing models fall short, research efforts (within the ASC Program, in collaboration with other funding sources such as Laboratory Directed Research and Development (LDRD) or DOE/Office of Science (SC) and in close collaboration with academic partners) can help identify possible extensions to existing models that can cover these gaps without having to switch to a completely new model. Further, where such extensions are not possible or impractical, additional research is needed to retrofit existing models to allow for a clean integration of new abstractions into existing systems. Coordinating this effort through the tri-lab would make this effort more impactful.	

CONCERN: Lack of Consensus in New Programming Models

The next decade is expected to be a period rife with uncertainty in regard to programming models and evolution of system integrated runtimes. Today, the traditional use of MPI, which is often bulk synchronous parallel (BSP)-oriented, is still the core model with some movement toward an MPI + X model with varying uses of OpenMP. However, the hardware architectures, scale, and data requirements driving toward exascale have motivated the development of newer (AMT) models based on asynchronous task and data execution that provide a more balanced approach to use of these environments, both at the node level and globally. However, the ASC Program is still in the exploratory stage whereby best practices are emerging; this is to be expected and is consistent with the long-term maturity cycle. The current wide range of approaches and programming systems being developed is a direct result of the immaturity of the field where best practices and standards have yet to be defined. While this diversity is natural and actually a good development since it helps explore the entire space, it prohibits long-term adaption due to missing consolidation and inability for risk mitigation through alternate implementations.

Strategy	Establish closer ties with programming model research efforts and increase early evaluation efforts.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Medium
ASC CSSE teams should continue research and integration efforts driven by current co-design, PSAAP, DOE/SC X-Stack and ATDM efforts and should integrate code development teams into the process to further understanding on both sides. Teams should seek closer collaborations with the development teams of the new programming models, both to convey ASC requirements to be integrated into the models and to help early evaluation efforts. Supporting CSSE efforts focused on investigating and developing the long-term capability required, such that the entire effort is not solely the responsibility of ATDM, are required. Examples of this would include the more exploratory components of that work, such as system-level load balancing and AMT-based resilience strategies. Additionally, progress on PGAS efforts should be tracked for relevance in complementary areas such as data analytics.	

Strategy	Support development of best practices and ultimately standardization.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Hard
For new programming models to be successful and be reliably available for long-term development efforts, such as for ASC nuclear weapons codes, standardization efforts are ultimately needed. While this is premature across the board, individual elements could be ready and the establishment of best practices can begin. ASC efforts should help identify these elements and help steer them towards a consensus across programming models and their associated run time systems. This could act as the basis for later standardization along open source reference implementations. Additionally, the ASC Program should help in the establishment of the necessary standardization bodies or help tie efforts to existing ones. The MPI process could function as a role model for these efforts.	

CONCERN: Transition to New Programming Models

Architectures are becoming more heterogeneous and diverse. Applications are complex with lifecycles that are tracked in decades. There will be issues in the transition process to move from traditional bulk synchronous approaches to the asynchronous approaches needed to effectively utilize architectures being developed. Currently, hardware is changing at a much faster rate than software can keep up with and the changes are dramatic. It is infeasible to transition entire software stacks or application codes to any new model, both due to the labor involved and the current immaturity of these models. Therefore, strategies are needed to enable incremental transitions where they are feasible and can provide the most impact for the program.

Strategy	Increase design studies, in close collaboration between system, library, and application developers.
Partners	DOE/SC, academia, PSAAP
Priority	Important
Difficulty	Medium due to coordination required for focused activities
ASC teams should undertake design studies that explore the integration of different programming models or systems within single applications. This should include application and library developers as well as system software designers. These studies should explore the extent of interoperability in the current software stack, document its limitations, develop new interoperability strategies, and share those with both runtime and application developers. Proxy applications, as developed for procurement suites and as part of the various co-design efforts, will play a key role for this strategy as they will provide the necessary experimental platforms.	

CONCERN: Missing Support for Multilevel Memory Models

One of biggest challenges to application performance are efficient management and use of data throughout the system. With new development of stacked memory and evolution of in-process capability, and varying levels of memory and data storage throughout the data environment, new models and runtimes will need tighter interaction with all aspects of the data system. These models must provide the right balance between transparent support of these new memory hierarchies to provide ease-of-use and exposure of memory systems (such as scratchpads or Non-Volatile Random Access Memory (NVRAM)) to enable targeted optimizations.

Strategy	Develop and test new application programming interfaces for memory management.
Partners	DOE/SC, academia, PSAAP
Priority	Important
Difficulty	Medium
New approaches are needed to help developers deal with the impact of new memory hierarchies. These have to range from implicit locality management to explicit memory management, for example, for scratchpad memories or NVRAM support. The ASC Program needs to ensure that these approaches are properly integrated into both programming models and runtime development. ASC needs to explore data management layers that enable significant elements of the data management (for example, system-wide data movement) to be handled directly by the RTS. These approaches and systems should consider the AMT programming model requirements as a key customer, as this is a critical enabling capability for AMT systems.	

Support for Co-Existence of Evolutionary and Revolutionary Runtimes

Along with developments in programming models and systems comes a complementary set of developments in runtime systems. While those are often hidden from the user (by the programming model), their properties and integration into the system have a substantial impact on how users interact with them and how they interact with each other.

CONCERN: Missing Runtime Interoperability

The most likely scenario is that no single programming model (and with that, no single runtime system) will be sufficient for all ASC code needs or that it will be impossible or impractical to unify codes onto a single, all-encompassing runtime system. For example, an application developed with one programming model may want to use a library developed with a different model. The ASC Program must therefore push for efficiently interoperable runtime environments. Note that this is a direct result of the maturity of new programming environments and the efforts in the ASC Program. The ASC Program is developing new codes and modernizing existing code but still needs compatibility between new efforts and existing code bases.

Strategy	Increase design studies, in close collaboration between system, library, and application developers.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Medium
ASC teams should focus on design studies that explore the integration of different runtime systems within single applications or within compound workflows that require interoperability between runtimes. This should include application and library developers as well as system software designers. These studies should explore the extent of interoperability in the current software stack, document its limitations, develop new interoperability strategies, and share those with both runtime and application developers. Proxy applications, as developed for procurement suites and as part of the various co-design efforts, will play a key role for this strategy as they will provide the necessary experimental platforms. Where necessary, results from this work should be fed back to runtime implementers and/or the associated standardization efforts.	

CONCERN: Lack of Mechanisms to Control Data, Thread, and Task Placement

Data, thread, and task placement is an increasingly important aspect of programming model and/or runtime systems. It not only affects the performance for accessing data but also has profound impact on power consumption and reliability. Current runtime systems often only provide rudimentary support for specifying data and thread placement, which is insufficient even for current platforms. Task placement mechanisms (for example, to support AMT models) are largely missing in current runtime systems. Further, these issues remain unsolved for multiple interacting runtime systems.

Strategy	Develop new intra-node data locality and movement abstractions.
Partners	DOE/SC, academia, PSAAP
Priority	Important
Difficulty	Medium
The ASC Program needs to experiment with APIs for explicitly programming threads, tasks, and memory. This can and must be aided by matching developments of production quality tools to automate code placement. The results from these experiments can and should be fed back to the respective runtime efforts to enable the development of more automated solutions as well as the establishment of cross runtime abstractions. Ultimately, such approaches should be transitioned into the respective standardization approaches.	

Strategy	Develop a new technique for inter-node data staging that is also applicable to composite workflows.
Partners	DOE/SC, academia, PSAAP
Priority	Important
Difficulty	Medium
Data locality and data movements are equally important in terms of off-node traffic, either in the form of I/O or for composite workflow (within applications, for example, mesher to solver, and to support in-situ analysis operations). The ASC Program needs to develop techniques to specify and implement such workflows in a way that they can be efficiently mapped to emerging architectures, for example, in support of novel parallel file systems, new non-portable operating system interface (POSIX) I/O abstractions, and new levels of the memory architecture such as NVRAM.	

Introspection across the Hardware/Software Stack

To support performance analysis, optimization, and debugging in such a complex, multilayer and composite software ecosystem, the ASC Program requires a new level of introspection in all components in the system. This includes the monitoring of resources, the observation of data transfers and data locality, as well as the tracking of execution units, for example, to monitor task balance. Additionally, administration and operations personnel must be able to track new metrics such as power/energy and reliability.

CONCERN: Missing Interfaces in the Hardware/Software Stack

Current software systems often do not provide the necessary public interfaces to export the necessary introspection information so that tools can extract the relevant data required for correctness verification debugging, performance analysis, or productivity tracking. This lack of interfaces often turns software systems into black boxes that defy any observation and ultimately the ability to optimize performance or show correctness. Further, even if interfaces exist, they are often ad-hoc, poorly documented, and change from version to version. This renders them mostly unusable for standardized tools and requires dedicated tool development for each version of each runtime element. This approach is not tractable, especially considering the expected increase in number of involved components as well as their anticipated rapid evolution over time.

Strategy	Survey existing interfaces.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Medium

Many programming-model and/or runtime systems contain at least internal or rudimentary introspection interfaces. In close collaboration with the developers of these systems, the ASC Program should create a list of existing approaches and document their similarities as well as their differences. Such a list can help define best practices, guide future interfaces, help establish more open interface "standards," and start the necessary dialog on how to optimize for new and upcoming programming models.

Strategy	Ensure tighter integration of introspection into the software stack.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Medium

The ASC Program needs a deep integration of introspection interfaces into programming models, runtime systems, and applications (through easy-to-use interfaces that can be permanently established in applications' code bases). This can no longer be an afterthought but must be established prior to (or at least as part of) initial application and system designs. Such interfaces must be useable at run time with minimal impact and support a wide range of tools, from performance analysis to auto-tuning, as well as for debugging, power, and resilience purposes. This capability should be available in all environments for AT and CT systems.

Strategy	Standardize introspection interfaces.
Partners	DOE/SC, academia, PSAAP
Priority	Highly Important
Difficulty	Medium

Ultimately, introspection interfaces must be part of the standard specification of each programming model, programming system, and runtime system. Standardization will allow the necessary introspection interfaces to be listed as requirements in DOE procurements. Recent examples for such efforts include the MPI_T interface in MPI 3.0 and the efforts around the OMPT/OMPD interfaces for OpenMP. Similar efforts must be established for other, newer programming models, preferably with similar interfaces to avoid an N:M support model for N models versus M tools. Industry partnerships such as the Power Insight system or the collaboration with Intel on power-aware HPC and the connected extensions to the Linux kernel, as well as academic collaborations with the development groups around Charm++, OCR, or Legion are examples.

CONCERN: Missing Integration of External Sensors

The performance of systems and applications increasingly relies on not only the HW/SW stack itself but also on external influences, such as provisioned power, machine room conditions, system configurations, and scheduler decisions. Data from these system elements must be included in any analysis and therefore require introspection interfaces as well, which must be accessible from the software stack.

Strategy	Provide introspection into job-level data through the resource managers.
Partners	HPC facility managers
Priority	Highly Important
Difficulty	Medium
The ASC Program needs introspection interfaces into resource and job management systems to allow both programming systems and runtime systems to understand the environment and to help inform decisions on future resource requests (for example, for elastic applications or large, dynamic ensemble calculations). Any new resource managers must include the necessary introspection interfaces.	

Strategy	Integrate and provide access to machine-room level sensors.
Partners	HPC facility managers
Priority	Highly Important
Difficulty	Medium
System performance will increasingly depend on machine room conditions, such as ambient temperature or available power, and the ASC Program needs continue to develop the necessary APIs to make such data accessible.	

Performance Tuning and Optimization Tools

Application development is in a state of flux. While current nuclear weapons codes are based on 20 years of development with a relatively stable programming model, architectures are changing radically to drive toward exascale capability. This change is causing disruption in code development strategies, posing new performance analysis and optimization challenges. Developers need the performance tools to aid them in performance analysis and evolving codes to take advantage of new architectures. These tools need to support *both* the evolving current approaches and the new approaches that are driving toward an asynchronous task-based programming model. The latter requires a new level of introspection combined with new metrics to help users explore the dynamic environments in which these task models operate.

The ASC Program requires the design and development of new, yet production-ready tool chains that help to develop and efficiently deploy performance analysis and optimization techniques. These tools will allow investigation of bottlenecks in all levels of the software stack as well as across a wide range of metrics, including resilience, power, and dynamic load balancing. These tools must be able to deal with the highly dynamic and asynchronous nature of next-generation systems and selectively hide or expose this adaptively from/to the user as necessary for the respective optimization goal.

CONCERN: Crosscutting Issue—the Need for Scalable Infrastructures

A key requirement toward exascale is system integration across software stacks, resources and hardware with communication and feedback loops. This has been noted before from tool developers but is becoming an important concern for next-generation systems due to the increasing complexity in both hardware and software stacks. As a consequence, the necessary infrastructure to collect performance at potentially high data frequencies, while maintaining efficiency at scale imposes substantial development efforts on this infrastructure. Best practices would be to develop a common scalable data aggregation infrastructure to support both application and system tools. It is infeasible to develop such infrastructures as one-off techniques separately for each tool.

Strategy	Develop tool-, system-, and application-agnostic tool infrastructure.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
The goal of a tool-, system-, and application-agnostic tool infrastructure is to enable the reuse of data collection mechanisms across tools and to efficiently target the integration of data sources across all available introspection interfaces. In addition to the actual tool measurement and observational data, this also includes meta- and/or context data on applications, programming models/systems, OS/communication, and hardware systems. This should also enable semantic correlation across all data sources. Such an infrastructure should be both measurement interface and tool agnostic and enable scalable data transport and aggregation.	

Strategy	Integrate tool infrastructures across components and make them available as a shared system service.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Hard
The need for a data-gathering infrastructure is not unique to tools but is shared with many other software components, such as visualization and data analysis, checkpointing and I/O, debugging, as well as with resource managers. As a consequence, research is necessary to investigate these commonalities with the goal of establishing a shared infrastructure with common APIs. Ultimately, such services should be provided as part of a distributed/parallel operating system.	

CONCERN: Limited Tool Support for Investigating Memory Usage and Efficiency

Data movement is the key to performance and power efficiency in upcoming architectures. Memory access behavior and bottleneck identification, as well as efficient usage of memory, will be critical to ensure efficient usage of next-generation machines. These memory systems are expected to be starved both in terms of amount of (fast) memory as well as bandwidth off chip (Memory Wall). However, understanding the limitations and their impact on application performance is currently difficult due to limited tool support in this area. If not resolved, this will have a critical impact in the ability to efficiently use those systems.

Strategy	Develop memory usage tools.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
The ASC Program needs to continue the development of tools that identify how much and how well memory is used and expand them to meet programmatic needs. In particular, their scalability needs to be enhanced, they need to allow a more direct correlation with the levels of the software stack causing the issue, and they need to be enhanced to deal with new layers in the memory hierarchy, such as High Bandwidth Memory, GPU memory, scratchpads, and NVRAM.	

Strategy	Develop tools to better understand node-local and job-wide data transfers.
Partners	DOE/SC, academia, industry
Priority	Important
Difficulty	Medium
The ASC Program needs to continue to evolve current tools that help with the tracking and attribution of data transfers to source code. This is mostly done through hardware counters and the program needs to involve vendors to ensure that future systems provide the necessary hardware information for this purpose. Additionally, new tracking techniques, such as Intel's PEBS or AMD's IBS, enable a new level of detail that is highly useful for developers, and tools are needed to extract, analyze, and display this information. Tools should be extended to support large-scale systems in a platform-independent way. Additionally, tools must be extended to support analysis for deep memory hierarchies and data movement.	

CONCERN: Tools Lacking for Asynchronous Task/Data Programming Models

New programming models, especially if they follow new paradigms such as asynchronous task-based models, will also require a new set of tools to help understand and eliminate the new types of bottlenecks these models expose. For this, the ASC Program will need to rethink the approach to assess performance and optimization as well as develop new metrics. A prerequisite for this is the availability of introspection capabilities and APIs, as discussed previously in this section. However, raw data availability alone will be insufficient without new analysis techniques.

Strategy	Develop tools and new metrics applying to task-based systems.
Partners	DOE/SC, academia, industry
Priority	Important
Difficulty	Medium
Task-based models provide a much more flexible way of scheduling computation (in the form of tasks) to processing resources. Existing tools, which in most cases assume an implicit static connection between computation and executing hardware, need to be extended to deal with these new models. Further, the ASC Program requires new metrics that can identify new bottlenecks, which can surface due to bad scheduling decisions or dependencies in task graphs. As part of this activity, an inventory of currently available low-level and internal tools by vendors and programming model developers is needed. Tools in this area will face two major hurdles specific to task-based systems: 1) the program needs tools that can tune one (possibly compound) application, even if this is based on multiple programming models; and 2) such tools must include aspects to deal with managing data placement, prefetching, and locality.	

Strategy	Develop tools to support porting to new programming models.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Hard
Porting to new programming models is currently difficult and time consuming. Therefore, the development of tools is needed that 1) help estimate the potential benefit in porting a code, and 2) help tune a task-based code to match the granularity in the system. The latter is a critical question that must be answered in order to reach good performance. Moreover, the evaluation of the code development environment requirements of an AMT-based system and a workable roadmap to incorporate those additional capabilities into ASC development environments are needed. Examples of this include debugging, monitoring, and performance tuning.	

CONCERN: Tools Lacking Support for New Hardware Features, including Accelerators

The upcoming Trinity and Sierra advanced technology systems will both feature heterogeneous system architectures (Intel KNL and NVIDIA GPU), and future architectures are expected to include similar heterogeneity. However, the programming environment for these accelerators is limited and good performance tools are still missing in many cases. Further, portability across different hardware architectures is complicated by a lack of consistency in the memory management and the low-level software layer (for example, accelerator access).

Strategy	Enhance current existing hardware and vendor-agnostic tools (possibly by integrating vendor-specific approaches).
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Hard
The ASC Program needs to enhance current HPC tools to provide support for these new hardware architectures. Many of these tools already provide rudimentary support and, working with the development teams, the ASC Program needs to expand these efforts to match the support with the details of the upcoming systems. Interactions with hardware vendors will be critical to establish the necessary interfaces and to integrate some of the existing proprietary tool technologies that vendors already have.	

Strategy	Develop tools to model and estimate benefits from porting to new architectures.
Partners	DOE/SC, academia, industry
Priority	Important
Difficulty	Hard
Porting to new programming models is difficult and time consuming. Therefore, tools need to be developed that 1) help estimate the potential benefit in porting a code, and 2) help tune a task-based code to match the granularity in the system. The latter is a critical question that must be answered in order to reach good performance. Existing simulators can serve as a basis for this, extended as needed to provide necessary analysis capability.	

Debugging and Correctness Tools

The changes in programming model not only impact performance but also the ability to debug code written for these new architectures. While many infrastructure issues are similar to the ones described in the performance tools section (and should be covered as shared infrastructure for performance and debugging tools), a few special concerns and strategies apply, which are described below.

CONCERN: Currently Limited Support for Next-Generation System Designs

While the ASC Program has a good debugging strategy across a wide range of tools for current systems, the anticipated systems will provide a more diverse environment for which coverage is still limited. This not only includes scaling (especially in terms of number of threads to debug) but also support for code off-loaded to accelerators and the support for new programming models, especially when deployed within complex software stacks.

Strategy	Extend existing debugging support for next-generation architectures and runtime systems.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Hard

The ASC Program has a solid base in debugging tools but needs continued support and evolution of these tools to support existing and evolving programming models and hardware architectures. These include scalability enhancements to tools for parallel debuggers, runtime extraction interfaces for OpenMP information, automating interactive debugging at scale, stack trace analysis tools, and outlier detection. Additionally, the program needs (in close collaboration with hardware vendors) to expand support for hardware accelerators as well as (in close collaboration with programming and runtime providers) address the debugging needs of new programming models, especially for task-based models. This has a close relationship to the need for proper introspection interfaces—efforts such as the OMPD debugging interface for OpenMP should be continued and expanded to other models.

Additionally, AMT models are expected to give rise to a new class of complex debugging challenges, resulting from the very nature of the (non-deterministic) computation. Understanding and reproducing the execution of a large-scale AMT-based application without the debugging (and tuning) tools needed to comprehend the behavior will be daunting if not practically impossible at some scale.

CONCERN: Pre-Emptive Correctness Checking Too Limited and Does Not Support Emerging Runtimes/Programming Models

Correctness tools, that is, tools that can identify bugs in codes before their actual execution or can detect otherwise hidden bugs at runtime, have proven to be highly useful. However, the currently supported range of features for existing programming models as well as the support for emerging programming models is limited and needs to be expanded.

Strategy	Enhance current existing hardware and vendor-agnostic tools (possibly by integrating vendor-specific approaches).
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium

The ASC Program must continue to improve existing tools that address scalability of MPI (+X) usage), reproducibility, and tools for hybrid static/dynamic check for data races. Additionally, the development of new debugging tools to support emerging asynchronous task-based programming models must be supported. The latter requires an in-depth evaluation of the semantics of the programming and the codification of these constraints.

Support For Efficient and Scalable Resilience

With the diversity of the hardware architectures and evolving software stacks, it is expected that support for efficient resilience will be critical and will have to be integrated into the applications and runtimes. While this is being developed, tools may have a supporting role to help develop the interfaces and approaches as well as developing specific tools that may serve as prototypes to integration or as a tool based capability. Additionally, the ASC Program needs to continue existing efforts, like high-performance I/O (HIO), scalable checkpoint restart (SCR) and local failure/local recovery (LFLR), which can serve as interim, if not long-term, resilience solutions.

CONCERN: Interfaces across the HPC Ecosystem Are Not Being Focused On and Currently Provide Limited Support

As for the runtime systems, resilience approaches will require direct access to new levels of the memory hierarchy, such as burst buffers or NVRAM. This requires a new set of interfaces and APIs. Additionally, expanding existing and emerging programming currently only provides resilience in a very rudimentary manner, if at all.

Strategy	Play an active role in the definition of new memory interfaces.
Partners	DOE/SC, academia, industry
Priority	Important
Difficulty	Medium
The utilization of NVRAM will be critical for efficient resilience techniques, as they allow short latency storage of I/O in general and checkpoint data in specific. The ASC Program must work with vendors and OS providers to establish the necessary interfaces to be able to integrate this resource into user-level and application integrated resilience techniques.	

Strategy	Play an active role in the inclusion of fault-tolerance (FT) techniques into runtimes and programming standards.
Partners	DOE/SC, LDRD, academia, industry
Priority	Highly Important
Difficulty	Hard
The ASC Program must play an active role in the integration and standardization of resilient programming models and help ensure that newly developed abstractions are both useful for current codes and meet the requirements of current and future nuclear weapons applications. This includes the continued work to develop the right fault tolerance abstractions in models such as MPI, the efficient support for new programming models by lower level runtime systems, and the support of new paradigms such as scale bridging codes or ensemble computations. Continued research is needed in developing and integrating fault tolerance capability into AMT systems. While these systems have known advantages in supporting certain FT strategies, the community is still exploring these approaches and is only now beginning to define best practices. A sustained effort is needed to mature this technology.	

CONCERN: Existing Resilience Approaches Will Not Scale to New Machines

While existing resilience approaches such as check-pointing restart have been shown to scale to current systems, their expandability to new architectures—which will be larger in terms of available parallelism, have deeper memory hierarchies and offer less favorable off-node bandwidths—is unclear. Nevertheless, they play a critical role in running applications codes efficiently.

Approaches such as localized failure and recovery show promise but are still at an early stage of the maturation cycle. Approaches such as these, which are provably scalable, need to continue to advance toward production-ready capability.

Strategy	Expand efforts in efficient checkpointing and integrate into overall software stack.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
Scalable checkpoint and restart techniques are critical and need to be available on any future machine. Core functionality needs to be extended to deal with larger degrees of parallelism, to support the utilization of Burst Buffers, and to integrate and optimize new techniques such as asynchronous I/O bleeding and checkpoint compression.	

Power-Aware and Power-Limited High Performance Computing

Power is expected to become one of the, if not the, most limiting factors as the ASC Program moves beyond the ATS-1 and ATS-2 machines. With a fixed power cap for an exascale machine, currently anticipated at 20 MW, the ASC Program needs to develop both algorithms and applications that require less power, but also that work well under externally imposed power constraints. The latter is important, since it is generally expected that power will no longer be provisioned to meet peak, but rather average demand, leading to the implementation of system as well as node wide power bounds, which can have drastic impacts on application performance.

CONCERN: Lack of Tools that Help Developers Understand and Influence Power Impacts of Their Design Decisions

Application and system software designers alike will be asked to implement more power efficient algorithms and approaches, as well as systems and applications that behave well under power-bounds. Yet, very limited tools and test environments currently exist that provide developers the feedback on power usage and impact necessary for the development process.

Strategy	Increase power instrumentation and access to power (and thermal) data in future machines.
Partners	Hardware vendors based on existing instrumentation experience
Priority	Important
Difficulty	Medium
While the ATS-1 and ATS-2 machines are not expected to be affected by power caps, they are likely the last generation of machines without these limitations. They need to serve as full-scale test beds for future application and system software developments and, therefore, need to be outfitted with more power instrumentation features as well as the respective APIs to gather such data from a system as well as an application point of view.	

Strategy	Provide new generation of tools that enable the correlation of power, power caps, and performance.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
The ASC Program needs a new generation of tools that 1) enables developers to understand the power impact of their design decisions and 2) provides the necessary emulation environments to test their software in power-constraint environments (for example, by artificially setting power caps or controlling thermal environments). Such tools should be user focused for interactive use as well as be available within automated regression environments. This strategy should be closely aligned with the performance tool strategies. Projects such as the SST simulator can serve as a basis for this, extended as needed to provide necessary power analysis capability.	

Strategy	Develop high-level abstractions to hide the impact of power-aware programming.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
Developing power-aware software is difficult and should not be left to the application developer. Novel techniques are required to give developers high-level abstractions, for example, in the form of annotations that can be used by compilers and runtime systems to infer power usage and power optimization opportunities without having the user specify them explicitly. Examples can range from simply marking phases and phase transitions to annotating expected compute or memory intensity, that is, properties that are well known to developers yet can help systems anticipate power usage.	

CONCERN: No Integrated Approach for System-Wide Power Caps and the Impact on System Design

Once systems provision power at average rates rather than peak rates (as anticipated), the ASC Program must provide techniques that allow guarantee of power caps across various levels of the system hierarchy (for example, system, rack, board, node, and core). While doing this at the processor level and below can be left to hardware (for example, through techniques such as Intel's RAPL) and will be the responsibility of the hardware vendor, power capping and distribution across sockets and nodes will have to be software controlled, which opens up wide security and deployment concerns.

Strategy	Develop power management system and policy.
Partners	Industry, OS developers, facility managers
Priority	Highly Important
Difficulty	Medium
The development of interfaces and tools with a focus towards system-wide runtime management of power and secure power distribution schemes that can be offered as system services is needed. This is both a technical and a political issue, since the program is transitioning safe system operation from hardware-only control to a hybrid hardware/software solution, potentially even based on in-house software systems. Experiments on current systems and analysis of system response to power caps will inform these policies over the next few years.	

CONCERN: Power as a Constraint Leads to Performance Issues

Applications on power-capped systems will face more than just a linear slowdown from less available power. The impact of power capping on application performance will depend on the algorithm and its utilization of resources, such as logical units and the memory system as well as properties of the individual processors caused by manufacturing differences. Currently, the ASC Program has a limited understanding of these performance issues, has limited support for predicting performance under power caps, and has rudimentary support to provide runtime optimization under such constraints.

Strategy	Develop performance models for applications under power caps.
Partners	DOE/SC, academia, industry
Priority	Highly Important
Difficulty	Medium
Modeling efforts to understand the performance implications of running applications and system components, as well as composite workflows (for example, for in-situ analysis under power caps) are needed. This will be essential to optimize performance of individual application elements and to allow schedulers at various levels of the software stack to direct power to where it is needed, thus ensuring the highest performance and/or throughput.	

Strategy	Expand efforts in the development of power-aware runtime scheduling and integration into other runtime efforts.
Partners	DOE/SC, academia, industry
Priority	Important
Difficulty	Medium
The ASC Program needs investments in runtime infrastructures and tools that (online) assess application power requirements and use this information to balance and optimize resource use. This should be done in a coordinated fashion at all levels of the system hierarchy, from local task management in low-level runtimes to system-wide scheduling and dynamic load balancing. Such systems should further help balance, and where possible even exploit, processor inhomogeneity, direct resources to the critical path, and identify optimal configurations.	

System Management and Monitoring

Introduction and Background

Energy demands for exascale systems will stress any facility. Interactions between the system itself and the services provided by the facility for power, cooling, scheduling, maintenance, repair, and updates will need to be more tightly integrated than in today's environment. This reality places a higher importance on monitoring of platforms, networks (both local and wide area), data storage systems, computer facilities, water supply, and energy delivery. The impact on an electrical provider of a sudden drop in demand of tens of megawatts should a full-scale compute job fail is a significant new failure mode not seen in the previous scale of computers.

In addition, extreme-scale HPC system procurements require a close partnership and a high reliance on the system vendors. This is especially true for system management and monitoring. Given that, the approach of this technical area is to consider ASC Program experiences with past systems (for example, Roadrunner, Cielo, Purple, and Sequoia). In particular, what does the ASC Program want HPC vendors to understand regarding how the program will manage and monitor systems as it moves toward exascale?

System management for ASC includes adding/updating software across the entire system, modifying the configuration of the system, scalable booting of the entire system, scheduling and monitoring jobs running across the system, and integration into the site file systems. System management is grouped into the following development/deployment themes: configuration management, hardware management, and software management. Monitoring for ASC includes concerns related to management as well as platform and application efficiency.

As noted in the Programming Environment and Tools technical area, a key requirement toward system management and monitoring at exascale will be system integration across software stacks, resources, and hardware with communication and feedback loops. As a consequence, the necessary infrastructure to collect relevant information at potentially high data frequencies while maintaining efficiency at scale imposes substantial development efforts on this infrastructure. This will require integration of external sensors and missing interfaces in the hardware/software stack. Due to these requirements, there is overlap in some of the Concerns/Strategies between the Programming Environment and Tools technical area and the System Management and Monitoring technical area. Best practices would be to develop a common scalable data gathering and aggregation infrastructure to support both technical areas. It is infeasible to develop such infrastructures as one-off techniques separately for each area.

Configuration Management

Given that the individual component count of pre-exascale systems is expected to be quite large, all aspects of configuration management must scale to accommodate these larger counts.

CONCERN: Vendors Develop/Test at a Much Smaller Scale

The cost of pre-exascale systems may present challenges for vendors when it comes to in-house, at-scale testing. Vendors should consider scale from the very early stages of design and development whether or not they have the resources to fully test at scale.

Strategy	Co-design system management tools using tri-lab experience.
Partners	Industry (including integrators and technology providers)
Priority	Highly Important
Difficulty	Medium
At a minimum, a means to simulate scale for development/testing is highly desirable. The tri-labs have experience managing large-scale systems; therefore, early involvement by tri-lab staff with design/development decisions is paramount.	

CONCERN: Configuration Management Not Amenable to Automation

More consideration needs to be given to the amount of human interaction required for base installation of pre-exascale systems as well as ongoing maintenance activities, including field notices, patches, and software updates. It is not uncommon for base installations and/or updates to involve monolithic scripts with complex failure modes, and the software may be delivered in a multitude of formats such as tarballs to be extracted, binaries to drop-in, and RPMs that may change their install behavior based upon what environment variables are set.

Strategy	Utilize industry standard best practices for configuration management.
Partners	Industry (including network, storage, and infrastructure), OpenSource community, standards bodies, academia
Priority	Highly Important
Difficulty	Medium
Vendors should supply a configuration API that allows for automated system software installation, including installation by existing site automation tools. Software updates should be supplied using industry standard best practices and packaging appropriate to the distribution. Full-system installations should be reproducible from a known base state. Large-scale systems have many hardware components that are managed independently (even though they are coming from the same vendor). To successfully accomplish efficient configuration management in this situation, the system administrators should have a single tool capable of managing all components and their dependencies.	

CONCERN: Lack of Reliable/Parallel In and Out-of-Band Communication to All Components

To successfully accomplish efficient configuration management of a pre-exascale system, system administrators have to have confidence that communications to all system components are reliable and can be done in parallel. System commands must be scalable up to every node in the system or to a subset of nodes. Reliable communication is extremely important to ensure whatever command is issued gets executed as expected. If there is any hint that a command has failed, then the system administrator will have to circle back around to the suspect component, which is a highly undesirable workflow at scale.

Strategy	Provide a fast and reliable interface to all system components and implement mechanisms for recording communication failures.
Partners	Industry (for example, component manufacturers)
Priority	Essential
Difficulty	Hard
In the case where there is a communication failure, one solution would be to implement a mechanism for recording the failure in a database that the system administrator can query to see what components need extra attention. There must exist a fast, reliable interface to query to ensure any given component in the system is configured correctly. This includes firmware versions and configuration as well as software package versions in addition to any configurations associated with them.	

CONCERN: Need to Accommodate Rolling System Updates

It will be prohibitive to schedule a full-system downtime to update system software and/or firmware. For pre-exascale systems, it is desirable to update software and/or firmware on a subset of the system at a time. This would reduce the impact to users and allow for testing of new software and/or firmware releases on the target hardware at scale. In this case, the system environment must allow for inconsistent software/firmware across system components and ensure compatibility between consecutive releases.

Strategy	Vendor must support rolling system updates.
Partners	Industry
Priority	Highly Important
Difficulty	Medium
Vendor must support multiple versions running simultaneously across the system in support of rolling updates for both hardware and software.	

CONCERN: Timely Full-System Restart

While the ASC Program needs the ability to do rolling system updates, there will also be times when a full system restart is needed. Restarting the entire system must happen in a timely manner. Often this is not tested at scale by the vendor until after the system is delivered. Short-sighted design of large-scale systems can make bringing up a full system at one time impossible and may require staging the booting of the system, thus taking many hours to days to get a system back into full service. Vendors need to think big—not just about what they can do in their lab with one node.

Strategy	Provide a hierarchy of service nodes.
Partners	Industry
Priority	Essential
Difficulty	Medium
A hierarchy of service nodes to address system startup scaling issues instead of relying upon a single service node is the desired solution. There should not be any single point of failure; in other words, no one component should prevent any other component from booting.	

Hardware Management

As with configuration management, hardware management must account for the large number of individual components associated with pre-exascale systems when considering how to effectively and efficiently manage the hardware of such systems.

CONCERN: Lack of Reliable Hardware Responsiveness to Requests

The component count of pre-exascale systems necessitates that hardware reliably respond to requests (for example, instantaneous node power off). When the hardware is asked to do something, it must do it. Currently, the operations teams have to track what does not work as expected and deal with it after a maintenance period is complete. Again, this is a highly undesirable workflow at scale.

Strategy	Provide a fast and reliable interface to all components of system and implement mechanism for recording communication failures.
Partners	Industry
Priority	Essential
Difficulty	Hard
The management of and communication to hardware must be reliable. The ASC Program needs a scalable database that tracks the state transition of components to reduce wasted time repeating tasks on non-responsive hardware. Such information could be combined with the configuration information requirements.	

CONCERN: Lack of Hardware Management Modularity

System administrators need a way to isolate parts of the system and debug those parts without impacting the entire system. Hardware should be modular such that parts of the system can be usable while other parts are unavailable, including for testing.

Strategy	Provide capabilities for isolating failed hardware components.
Partners	Industry, software tool companies
Priority	Essential
Difficulty	Medium
The idea of a system scalable unit currently used by the capacity Linux clusters could help address this concern. If isolation is not possible, then a well-documented procedure for debugging the potentially failed hardware is essential.	

Software Management

While most of the system management concerns fall under configuration management, it is important to consider the impact of system scale on software management as well.

CONCERN: Management of Programming Environment at Scale

Local programming environment software (for example, scientific libraries and compilers) must be provided in a scalable way. The software must be synchronized across nodes of a system and it must be updateable without system downtime.

Strategy	Provide a scalable software management solution.
Partners	Industry
Priority	Highly Important
Difficulty	Medium
A scalable software management solution is needed for both system software and vendor or locally developed software. This may require a reliable hierarchy or way to install locally into images without taking up too much memory. The solution should be compatible with other scalable software distribution methods used on the existing systems.	

Monitoring

Monitoring, analysis, feedback tools, and infrastructure are needed to enable pre-exascale HPC. Efficiently utilizing platform resources in terascale and petascale environments has become increasingly challenging due to a lack of information about how applications are utilizing resources across the large numbers of components being allocated to them. ("If you don't know what is happening you can't make it more efficient.") Even a gross understanding of a large-scale application's behavioral characteristics over its lifetime with respect to memory occupancy and bandwidth is typically lacking. Additionally as more applications and processes per application concurrently contend for shared resources, congestion further degrades performance. Runtime variations of 50% or more have been observed and attributed to contention for shared network resources on modern petascale platforms.

While more intelligent scheduling, resource allocation, and task mapping could be used to mitigate many of these problems, all of these solutions rely on information (for example, an application typical memory, CPU, network, and file system resource needs) that currently is not collected. To gain such information, much more detailed run-time system-wide, application-aware monitoring is required. Additionally, analysis tools will need to be developed for trending, run-time decision making, and feedback to all levels of the stack, including applications. These tools will need to process and correlate information from both numeric and text logs.

The risk in this area is that applications will not perform sufficiently to solve mission-critical problems in the time available.

CONCERN: Without Active Monitoring and Feedback, Load Balance Issues Will Be an Impediment to Application Scaling

It is recognized that good load balancing of processes across large-scale applications is crucial to performance and scalability. As systems become larger and more heterogeneous, it will become more difficult, if not impossible, for an application to maintain load balance across its many processes due to run-time variability in access to shared resources caused by competing applications (for example, network, parallel file systems, and burst buffers). The dynamic use of power limiting technologies will further exacerbate the problem.

Strategy	Develop scalable automated tools to quantify behavioral characteristics of applications.
Partners	Industry, academia, CSSE
Priority	Highly Important
Difficulty	Hard
Scalable automated tools are required to quantify the behavioral characteristics (including system impact) and their ranges for particular applications, including run-time parameters with respect to the shared resources they utilize. Further, tools to utilize these behavioral characterizations to influence scheduling and resource allocation decisions to minimize their adverse effects are needed. These characterizations can also be utilized to detect abnormal behaviors that may indicate resource contention, imbalance, and performance degradation due to component degradation/failure.	

CONCERN: Single Component Degradation/Failure Can Result in Large-Scale Performance Impact

Degradation in performance of individual components due to ageing effects, power management, and hardware/software failure can significantly degrade application-wide or even system-wide performance. For example, failure in a valve component on a water-cooled system may result in thermally throttled CPUs, which would then be the bottleneck for all the processes of an application—even those not running directly on the affected components. This would introduce an effective load imbalance.

Strategy	Develop scalable monitoring and automated analysis tools to detect known degradation/failure modes.
Partners	Industry, CSSE
Priority	Highly Important
Difficulty	Hard
Scalable monitoring and automated analysis tools to detect known degradation/failure modes either directly or via symptomatic behaviors, to analyze the scope of adverse effects, and to interface with schedulers, resource managers, and applications are needed. Further, these tools should notify administrative staff of the problem(s). Note that this also depends on well-defined APIs. While the minimum level of effort here would be to address known degradation/failure modes, the ideal would be that these tools also help to discover root causes and correlative relationships between root degradation/failure and symptoms, including change in behavioral characteristics of numeric data being monitored and log file information.	

CONCERN: Large-Scale Machines Will Require Resource and Application-Aware Scheduling and Resource Allocation

It is well known that contention for shared resources can account for large application performance variation even on today's systems. Contention can result in an imbalance across an application's processes or in delays of all processes, for example, if all processes are blocked due to file I/O. To minimize such contention, scheduling and resource allocation schemes will require suitable information about the application's resource needs as well as the current resource utilization of the system and how this profile is likely to evolve given the running applications.

Strategy	Develop scalable automated tools to quantify behavioral characteristics of applications.
Partners	Industry, CSSE
Priority	Highly Important
Difficulty	Hard
The ASC Program must develop scalable monitoring and automated analysis tools to detect known degradation/failure modes either directly or via symptomatic behaviors, to analyze the scope of adverse effects, and to interface with schedulers, resource managers, and even directly with applications to minimize impact. Further, these tools should notify administrative staff of the problem(s). Note that this also depends on the existence of a scalable monitoring infrastructure. While the minimum level of effort here would be to address known degradation/failure modes, the ideal would be that these tools also help to discover root causes and correlative relationships between root degradation/failure and symptoms, including change in behavioral characteristics of numeric data being monitored and log file information.	

CONCERN: Mismatch between Application Needs and Platform Resources Can Result in Low Platform Efficiency and Applications Not Able to Perform Timely Executions

While system architects strive to provide a platform that meets the needs of the normal workload expected to be run, the needs are often based on simulation and rules of thumb rather than hard data on actual utilization and bottlenecks on previous platforms. Understanding the real bottlenecks, contention, and failure modes on today's platforms can better inform future designs.

Strategy	Utilize behavioral characteristics of the application mix during procurement process.
Partners	Tri-lab CSSE, industry tool vendors
Priority	Highly Important
Difficulty	Medium
Next-generation system architecture should utilize actual behavioral characteristics of the application mix on current platforms to identify areas for improvement in future procurements. These can include identification of areas of either over or under provisioning. Over provisioning, while it will not directly adversely impact performance, means monetary resources were spent in the wrong place and the under provisioned resources will impact aggregate platform performance. Thus, matching platform resources to actual workload requirements as much as possible will yield the best overall performance.	

CONCERN: Inefficient Operation Can Result from Poor Problem Decomposition on the Part of a User Due to Lack of User Understanding of Application Resource Needs

Users do not necessarily have a detailed understanding of the resource demands of the application's they run (especially when they are not the code authors). As users move to new platforms/architectures and/or the platform components characteristics have changed substantially, this problem will only get worse. This can result in very inefficient use of platform resources, resulting in performance degradation system wide (for example, a user utilizing a single Lustre object storage target (OST) for heavy file I/O rather than balancing the load across an appropriate number can adversely impact the performance of other applications sharing that OST for their file I/O, even though they may be striping across an appropriate number of OSTs).

Strategy	Develop scalable monitoring/analysis tools to identify inefficient use of resources.
Partners	Industry, academia, CSSE
Priority	Highly Important
Difficulty	Hard
The ASC Program needs to develop scalable monitoring and analysis tools that run continuously as a system service and provide automated feedback to users and administrators when there are strong indicators of inefficient use of resources. This is not targeted as a control mechanism but rather as a method to identify possible configuration inefficiencies. Additionally, feedback from the user as to the accuracy of the feedback would enable refinement of the analysis methodologies over time.	

CONCERN: Similar Components Can Have Significantly Different Power Draw under the Same Workload while Providing the Same Performance

Similar components (for example, CPUs) can have significantly different profiles with respect to power draw and thermal properties while under the same workload and producing similar performance results. These differences may change with age.

Strategy	Develop scalable component power monitoring and profiling tools.
Partners	Industry, academia
Priority	Highly Important
Difficulty	Medium
Scalable component power monitoring and profiling tools, including a reference profile store, for use in allocation and energy management decisions are needed. Given power constraints and the fact that similar computational components can have significantly different power/performance characteristics (which may change with age), understanding these characteristics on a per-component basis and utilizing them in scheduling/resource allocation decisions can be significant in maintaining performance and staying within a power budget.	

CONCERN: Naive Automated Power Management Could Have Serious Consequences

ASC Program facilities are reaching power limits with respect to power and cooling on a subsystem level (for example, compute node), system level (for example, Trinity), and on a facilities level (for example, in the Livermore Computing Complex) as component densities and subsystem complexities increase. Thus, an understanding of workload effects, component variability, throttling controls, and throttling effects must be used in concert to effectively manage workloads' power needs. Note that even a sudden large drop in power draw from a large system could impact power quality to a whole facility. Understanding the potential impact to the facility of a full-scale job suddenly stopping in mid-run is critical to the operations teams and facility managers.

Strategy	Develop scalable power monitoring and profiling tools for computational components, applications, and facilities.
Partners	Tri-lab FOUS, tri-lab CSSE
Priority	Highly Important
Difficulty	Medium
Continuous system- and facilities-wide synchronized collection of right-fidelity power/energy data needs to be performed to understand application power profiles and their effects on both a system and facilities level. Further, this will enable system component mappings between workload and power draw, including same component variability and ageing effects.	

Strategy	Develop scheduling and resource allocation strategies to utilize component and application power profiles in scheduling, allocation, and power management decisions.
Partners	Academia, CSSE
Priority	Highly Important
Difficulty	Hard
Power profiles should be used in conjunction with effective power capping and frequency adjustment. Similar components may have substantially different power profiles given identical workloads and performance characteristics, and these may change over time due to ageing effects. Thus, when developing application power profiles, it is important to apply appropriate weighting factors based on the power/performance profiles of the components being used. Additionally, intelligent scheduler and resource allocation strategies should ensure incorporation of appropriate power profile characteristics of applications and system components.	

CONCERN: System Internal Power Estimates May Be Grossly Different than Physical Plant Realities

Power measurements made internal to a system may not sum to the actual power draw on a facility. This mismatch can come from a variety of sources (not everything is instrumented, sampling rates are low, etc.)

Strategy	Develop tools to perform power profiling of HPC systems and their application workloads at a physical plant level.
Partners	Tri-lab FOUS, tri-lab CSSE
Priority	Highly Important
Difficulty	Easy
Tools are needed to perform power profiling of application workloads at a physical plant-level as well as tools to enable use of these profiles by scheduling, resource management, and platform power management facilities. When HPC platforms have to be actively controlled to stay within some power draw envelope due to cost or power feed limits, ultimately what counts is the physical plant level power draw. Active power management is performed currently based on a platform perspective of power draw. Either analyses of the relationships between platform and facilities needs to be incorporated in the active power management system or the power management system needs to monitor the physical plant power draw with sufficient fidelity to ensure compliance.	

CONCERN: Accessibility of Error/Warning Messages of Components from Log Files

Valuable information contained in log files that could help in informing applications, schedulers, and resource managers is not being utilized because of information overload (typically, too much non-essential information to sift through). Root cause analysis is extremely difficult given the variety of components and volume of error/warning messages. Systems are in a constant state of error. It is difficult to properly identify significant versus insignificant error states, making the problem worse. Often the log streams are filled with fatal messages for known bad or disabled equipment. There are currently no viable automated mechanisms for harvesting information from system log files (for example, event, syslog, and console) and turning it into actionable information. These logs contain valuable information that, taken as an aggregate, can identify problems of contention, degradation, and failure.

Strategy	Develop tools for automated processing of all available logs.
Partners	Commercial software vendors, academia, or preferably open source solutions
Priority	Highly Important
Difficulty	Hard
Tools are needed for automated processing of all available logs, as an aggregate and including numeric monitored data, to: 1) harvest errors, 2) identify failing components, 3) provide meaningful remediation strategies using a configurable problem-solving engine, and 4) provide feedback mechanisms that can proactively deliver appropriate messages to system components and/or be queried directly for information. Active analysis of log files (for example, syslog, console logs, event logs, and scheduler logs), perhaps in conjunction with numeric data, will grow in importance due to the need for expeditious identification and mitigation of faults and their root causes. Currently, the majority of log files are typically aggregated to a single management host. The volume of log data will continue to increase with host and subsystem counts. This, coupled with the need for "real-time" processing of the messages, will make a flexible and distributed solution necessary.	

CONCERN: Lack of Well Defined APIs for Interaction between Monitoring and Analysis Services and Applications and Other System Services Will Cripple the Effectiveness of Scalable Monitoring and Analysis Tools Being Developed

Strategy	Develop an extensible set of APIs targeted at enabling external interaction with monitoring and analysis services.
Partners	Commercial software vendors, academia, or preferably open source solutions
Priority	Highly Important
Difficulty	Hard
While high-fidelity monitoring, scalable transport and storage, and run-time analysis will all be required to take advantage of larger scale machines, these are insufficient by themselves. To be effective, the results of run-time analyses will need to be conveyed to appropriate components of the HPC eco-system (for example, system software, system administrators, and applications) for appropriate action to be taken in a timely fashion. This implies the need for development of interfaces appropriate for interaction with each of these components.	

CONCERN: Current Data Sampling, Transport, Aggregation, Analysis, and Long-Term Management Used on Today's Systems Will Not Scale to Meet the Needs of the Pre-Exascale Environments

Today's petascale HPC systems employ either vendor/platform proprietary monitoring systems or go largely unmonitored. In the cases where continuous monitoring is employed, the data generally is used for health checking and is overwritten by the next sample. Where data is saved, it is typically a small subset of what is being sampled, is coarse grained (for example, one-minute sample periods), and is rarely analyzed. For pre-exascale systems to be effectively managed, they will need to be monitored at higher fidelities. As individual compute nodes host more resources and platforms become more heterogeneous (for example, specialized nodes such as burst buffers, and visualization) the number and variety of the data (metrics) will also grow.

Strategy	Develop scalable, platform-independent data sampling tools and methodologies that provide time synchronous samples across all monitored objects to produce a system "snapshot" capability.
Partners	Tri-labs FOUS, tri-labs CSSE
Priority	Highly Important
Difficulty	Easy
Develop platform-independent tools, methodologies, and infrastructure for HPC data collection that can be grown, as systems continue to evolve, to meet the ASC Program's needs with respect to effective scaling of both machines and applications through high-fidelity monitoring, run-time analysis, and feedback.	

Strategy	Develop scalable parallel storage and retrieval capabilities for sampled data where both in-band and out-of-band data are stored to targets accessible by analysis tools.
Partners	Commercial vendors
Priority	Highly Important
Difficulty	Easy
The required storage methods that must be developed are specific to HPC monitoring. Analysis as to longevity of which types of information (for example, log, numeric) from what origins (for example, syslog, console logs, and power) and if/how they should be pruned will need to be performed. Further, the characteristics of "hot," "warm," and "cold" storage with respect to transition triggers and methodologies will need to be determined. With respect to performance, the storage must be able to handle access from multiple requestors simultaneously querying large amounts of data over long periods of time. Additionally, data requestors will also want to process small segments of data from chunks of data going back many years. The higher fidelity of sampled data combined with the increased component count and increased instrumentation imply much larger storage systems will be needed. It is expected that every component of the overall system will provide multiple data points of interest, which will need to be collected.	

Strategy	Develop scalable analysis tools that utilize both numeric and log file data to perform both run-time and post-run analyses and expose results via an appropriate API.
Partners	Tri-labs FOUS, tri-labs CSSE
Priority	Highly Important
Difficulty	Medium
Multiple teams working with a cluster require multiple but often different sets of data points and/or views not always coming from the same data source. Some tools present data in text-based log files while others make numeric data available via files of some API. Tools must be developed that can work with both types of data—both separately and in conjunction—in both a run-tine and post-run fashion. A standard API that enables sharing of run-time information between analysis tools, applications, system software, and system hardware will facilitate more efficient system and application operation.	

Strategy	Develop and/or deploy scalable compression techniques for moving old data to long-term "cold" storage. This data needs to be retrievable for historic analysis purposes. Dedicate appropriate long-term storage resources to maintain this data at least over the life of the machine (~100 PB or more uncompressed).
Partners	Tri-labs FOUS, tri-labs CSSE
Priority	Highly Important
Difficulty	Medium
The volume of monitored data from pre-exascale systems is expected to be multiple terabytes per day. In order to be able to analyze how changes in application code, system software, and system hardware as well as unexpected events such as lightning strikes, chiller failures, noise on the feed power change system behaviors (including failure rates), the ASC Program wants to retain data for the life of a machine and beyond. Without an excellent compression algorithm and dedicated storage resources, the stored data will become unmanageable, thus resulting in the need to delete existing data. Such deletions are undesirable, as there frequently arises the need to query historical data from three to five years back. If historical data is aggregated, then detailed analysis will be adversely impacted or rendered useless.	

Data Analysis and Visualization

Introduction and Background

The central task for ASC data analysis is to develop effective technology that enables reduction, visualization, and exploratory analysis of extremely large simulation data. Growth in data scale, changes in system architecture, and evolution of system design due to economic limitations all result in continuing challenges for delivering this capability. As the ASC Program entered into the present era of petascale computing, the tried and tested methods of distributed post-processing developed under prior ASC research became constrained by the imbalance in performance growth between system compute and system I/O. This trend is likely to continue unabated.

The past several years have seen fervent work across the program to introduce new analysis and visualization methods that are less reliant on simulation state dumps and are more tightly integrated with simulation processes. This approach continues to be a promising thread of development for enabling visualization at future scales; however, it brings with it significant implications for data management and user work flow. Efficient in-situ image production employed on next-generation compute platforms will allow for a proliferation of analysis products. The management and utilization of these data streams is an impending, frontier area of research that will need to be addressed and properly supported by the program.

Further, the ASC Program expects that total scientific workflow on these systems will continue to increase in complexity. For full-system simulations, the arduous task of problem setup will soon require additional analysis design and data archival considerations. For ensembles or parameter studies, future systems will support a far greater volume of simulation runs, each potentially producing in situ analysis products. A greater degree of integration will be required among the analysis, simulation, job management, and data management tools that compose the program's modern supercomputing capability.

Despite these many promising research and development (R&D) areas, the ASC Program must continue to maintain present capability, both as a risk mitigation strategy and to facilitate a smooth transition away from legacy analysis techniques. The next five years will likely present fundamentally new compute and file-system architectures. The ASC Program should ensure it properly accommodates the software modification and development efforts that will be necessary to maintain compatibility of existing visualization and analysis tools.

The challenges of pre-exascale data analysis are faced by a range of institutions. The needs of these institutions are not identical, as they are necessarily tied to applications, research domains, and resource constraints. However, an important strategy for achieving an effective data analysis environment will be to leverage existing R&D efforts by partnering with DOE SC and other appropriate U.S. government agencies, industrial partners, and international collaborators (Atomic Weapons Establishment (AWE), Commissariat à l'Énergie Atomique (CEA)). Complementary goals on exascale data analysis are being pursued through the SC SciDAC's

Visualization and Analytics Center for Enabling Technologies (VACET) and SciDAC's Institute for Ultra-Scale Visualization; and the ASC Program should maintain existing ties to these centers to enable effective partnership.

The ASC Program has historically played an important role driving the analysis and visualization capabilities of the HPC community. As the program moves toward exascale, however, there are a number of emerging issues that the ASC analysis and visualization teams need to address to effectively serve the specific needs of the ASC applications and the broader needs of the general HPC scientific computing community. This section presents concerns and strategies to address four particular issues anticipated for pre-exascale environments: computational science workflows, algorithmic R&D for analysis and visualization, productization and deployment concerns, and cross-cutting challenges created by an increasingly complex and dynamically changing environment.

Computational Science Workflows

As the ASC Program looks ahead to next-generation architectures, the complexity of the workflows continues to increase. In particular, projected limitations in storage bandwidth and capacity are motivating development of in-situ analysis (in which analysis shares the primary simulation resources), in-transit analysis (in which data is transferred asynchronously to secondary resources), embedded uncertainty quantification (UQ), ensembles of runs to support V&V subprogram efforts, and integration of experimental and simulation results. However, the community currently lacks the tools to enable and facilitate these workflows at scale, both in research and production modes. Specific concerns and the strategies and investment areas needed to mitigate them follow.

CONCERN: Increasing Complexity of Workflows and Tools

As workflows become more complex, users will face difficulty in problem setup, scheduling, and data analysis. Users are likely to run problems at a number of scales as well as ensembles of simulations, which will require different data analysis methodologies. It is likely that combinations of legacy methods (disk I/O-based analysis) with newer methods, such as in-situ analytics, will be utilized.

Strategy	Develop methods that enable a combination of exploratory (human) analysis and in situ techniques.
Partners	CSSE
Priority	Highly Important
Difficulty	Medium
Post-hoc exploration of simulation artifacts is an important part of today's application workflow, but the increased reliance on in situ techniques limits the scientist's ability to perform exploratory analysis. While the potential amount of data generated is anticipated to increase by three-to-six orders of magnitude, the cognitive abilities of the human—also called the "human bottleneck"—remain roughly constant. Thus, scientists will not be able to explore large portions of the entire contents of the simulation data (for example, mesh). As a result, a number of recent efforts address this challenge by reducing the volume of datasets either through sampling (both spatial and temporal), compression, and feature extraction.	

Strategy	Develop tools to specify, modify, and deploy complex workflows in an efficient and intuitive manner.
Partners	CSSE, PSAAP
Priority	Highly Important
Difficulty	Hard
Future workflows will have multiple asynchronous data streams that users will need to combine and control; however, the ASC Program currently lacks the tools to specify and modify workflows in an efficient and intuitive manner. In addition to providing tools to construct and deploy workflows at scale, the community needs to invest in development of adaptive, data-driven control flow mechanisms to adaptively determine frequencies of I/O and/or expensive in-situ data analysis (rather than prescribing these a priori independent of simulation state).	

Strategy	Develop techniques to enable the use of partial ensemble analysis results to inform and direct job-management frameworks (for example, Dakota).
Partners	CSSE, vendors
Priority	Highly Important
Difficulty	Medium
The analysis of large ensemble results for uncertainty quantification, engineering design, and other purposes is rapidly becoming an important use case for HPC. Current methods perform all the analysis as a post-processing step after execution and completion of the, perhaps thousands, of jobs. Rather than wait for completion, it is desirable to develop ensemble analysis techniques that operate on partially complete ensembles, perhaps informing users and job management frameworks about trends that could be addressed through changes to the parameters of ensemble runs. Such capabilities could significantly improve resource utilization of large systems.	

Strategy	Develop standard methods for data management and provenance.
Partners	CSSE
Priority	Highly Important
Difficulty	Hard
Connecting simulation with experiments presents a set of data management issues, with facilities providing a changing set of experimental data and diagnostic models that must be tracked in addition to pre- and post-shot simulation results. The proliferation of complex workflows will strain current ad-hoc data management methods. Leveraging modern technologies to capture provenance and aid users in managing simulation setup, output, and analysis/visualization products is highly recommended.	

CONCERN: Software Complexity and Re-Use

In-situ and in-transit data analysis and visualization will require improvements in software construction to enable these technologies to be deployed widely. Deploying these algorithms will be made more difficult by a multiplicity of architectures with complex memory hierarchies.

Strategy	Invest in the creation of in situ components that can be leveraged by domain scientists for inclusion in their codes.
Partners	CSSE, PathForward
Priority	Important
Difficulty	Hard
Historically, simulation developers have implemented in-situ analysis by independently coding custom analysis routines in their simulation code-bases. This embedding of analysis methods has led to duplication of efforts across codes as fundamental data analysis methods (for example, geometry processing and level set extraction) are re-implemented in each code as they become needed. As advances in algorithms in the data analysis community occur, these embedded routines are often not improved to the state of the art. The goal of this investment is to develop data analysis components based on best practices in the data analysis community that form the foundation of both embedded analysis as well as loosely coupled in situ methodologies such as dynamically linked assets or co-processing schemes. Such components could be leveraged in an agile fashion across a variety of scientific domains and workflows.	

Strategy	Research in-transit techniques to leverage the increasingly deep memory hierarchies of future systems.
Partners	Tri-labs, industry partners
Priority	Important
Difficulty	Medium
Memory, storage, and archival hierarchies will continue to deepen as the exascale era approaches. Newly introduced levels in this hierarchy, such as the burst buffer, provide opportunities to execute out-of- and analysis processes without disrupting (or requiring dedicated) data movement or without impeding simulation work on the core system. Additionally, in-transit may allow for a right-sizing of parallel visualization processes or accommodate the introduction of data-intensive compute, or similar, resources that are optimized for analysis work loads. Accessibility and coordinated use of various levels of the storage hierarchy should be studied for these purposes.	

Algorithmic Research and Development

Due to the increasing disparity between disk I/O speed and compute speed, simulations do not retain high-fidelity data—because it is not possible to write out the breadth of information computed. This problem will be exacerbated with the move to exascale, and simply throwing away computed data is not an acceptable solution. Researchers and analysts need to retain as much information as possible from the exascale runs, so that further study of the information yields as much knowledge as possible, whether results are viewed in isolation or in comparison to other results. Lastly, if the program will support investigation, comparison, and quantitative analysis (including verification and validation, quantification of margins and uncertainties (QMU), UQ, and ensembles of runs) on exascale data, it will be necessary to write out rich information from these large runs. If this information is not available, much of the value of the computation will be lost to further study, thus devaluing the resource that most constrains compute time.

CONCERN: Combating the Data Movement Bottleneck while Maintaining Exploratory Capabilities

In situ and in-transit analyses provide a partial solution to the data movement problem. The ASC Program needs continued R&D investments for development of data analysis and visualization algorithms that scale. However, both in situ and in-transit frameworks require a priori knowledge of the questions one wants to ask of the data, and are thus limited to the study of anticipated phenomena. In many cases, initial results, in particular unexpected results, lead to new questions, calling for iterative exploration that must be done as a post-process. This motivates investments for in situ data reduction and management techniques to reduce the overall amount of data written to disk, while maintaining the key features of interest, thus minimizing the impact on subsequent analyses.

Strategy	Perform R&D for in situ data reduction.
Partners	CSSE
Priority	Important
Difficulty	Medium
Traditional compression techniques, and the in situ application of these techniques, to reduce the overall amount of data written to disk will remain an important challenge and development area. This will include both lossy and lossless techniques, depending on the use case. By compressing the data, not only is less data stored on disk, the effective I/O rates are improved because less data is being written and read from disk. Data must be decompressed prior to analysis, and therefore this approach alone is not expected to meet the anticipated increased storage needs at exascale.	

Strategy	Perform R&D for in situ anomaly detection and feature extraction.
Partners	CSSE
Priority	Important
Difficulty	Medium
Related to data compression, the ASC Program needs continued investment in 1) statistical feature extraction techniques that compute lower dimensional representation of high-dimensional data while still capturing the data with sufficient accuracy, and 2) segmentation-based feature extraction techniques that focus on the identification of relevant subsets of a spatial domain. Such representations capture the physics and relevant quantities of interest yet are typically orders of magnitude smaller than the raw data.	

Strategy	Perform R&D for data management and exploration.
Partners	CSSE
Priority	Important
Difficulty	Medium
Effective management of analysis and visualization byproducts requires the ability to organize and annotate the data so it can be read and searched as quickly as possible. This includes organizing the data so that all the data necessary can be read in one large chunk, the data is annotated so that minimal amount of data is read to perform the operation, and the data is organized so that approximate representations of the data can be quickly shown to the user for interactive processing. This effort should be strongly coordinated with the scalable I/O and system software communities. In particular, analysis and visualization are expected to be the key use cases that drive requirements and design of future storage systems.	

CONCERN: Uncertainty Quantification, Ensembles of Models, High Dimensional Spaces

Investigating a single petascale run is difficult; investigating hundreds or thousands of runs presents its own unique challenges, including unique data access patterns, I/O requirements, and the need for assisted discovery. The ASC Program needs methods for ensemble visual representations: abstract representations for both the ensemble itself and the analysis results. The representations may be different based the result data types and on the tasks being performed (for example, sensitivity analysis, parameter optimization, anomaly detection, uncertainty quantification, design space exploration, and algorithm comparison). Scalability limitations due to both limited screen real estate and human perception need to be evaluated (and perhaps different abstractions developed).

While expert practitioners leverage tools like R, the analysis and visualization language, to analyze these datasets and often apply complex algorithms in the process, it is likely that in the future domain scientists will be setting up, running, and analyzing large-scale ensemble data. Enabling a domain expert who is not simultaneously a trained statistician to understand (and trust!) his or her analysis is required to fully exploit the capabilities of next-generation architectures for these kinds of studies.

Strategy	Perform R&D for visual analytics of ensembles of runs.
Partners	CSSE
Priority	Highly Important
Difficulty	Hard
Simulation results must not be analyzed in isolation, so researchers must develop better methods of comparative analysis across simulation runs. This requires comparative techniques to be developed for a variety of data types (for example, multivariate scalar tables, time series,	

Strategy	Perform R&D for visual analytics of ensembles of runs.

graphs, meshes, images, tensor fields, and algorithmic models). Quantitative and qualitative comparison of different results is crucial to understanding the nature of differences between results. Assisted discovery of trends and features in the data will make it possible to understand large- and small-scale areas of interest between results. Moreover, the ability to "Google your data" for interesting features or having software that automatically identifies potential interesting features is crucial to helping analysts investigate and understand the data these platforms produce.

Certainly, a human can understand how varying a single variable can influence results, but optimization, V&V, and complex interplay between multiphysics codes can only be understood through harnessing visualization and analysis. This problem increases when the customer need to compare several results is addressed or when the V&V need to investigate ensembles of runs is addressed. Scientists need computational help to determine areas of interest within the data and to assist in exploring, analyzing, and annotating the data. Techniques viable at petascale will not suffice, simply because the quantity of data will outpace the ability of infrastructure to access the data.

Many of the existing tools are designed around small numbers of simulations in which users investigate a single simulation at sets of discrete time-dumps. UQ studies introduce the complexity of sampling parameters spaces in high dimensions and understanding the statistical properties of the resulting ensemble. Researchers and developers should integrate the ability to interact with these potentially huge ensembles with existing visualization and analysis capabilities where appropriate, and build new tools where needed to address limitations of existing tools.

Productization and Deployment

There are a number of fundamental R&D challenges that must be addressed for exascale architectures. However, without the necessary hardware investments and without a viable path to productization and adoption by the ASC and Directed Stockpile Work (DSW) analyst communities, the program will not reap the benefits of the new research capabilities that are developed.

CONCERN: Ensuring Viable Productization of Newly Developed Capabilities, and Suitable Usage Models for the End-User Community

Delivery of a viable pre-exascale analysis capability to the ASC and DSW user communities necessitates robustness, reliability, and only non-disruptive modifications to established workflows. These communities maintain a critical dependence on the validation of newly developed capabilities against a broad base of simulation and analysis results acquired over the full course of their respective histories. Only by accommodating this validation process and demonstrating functional reliability will any new capabilities be efficiently adopted. Additionally, ASC must take into consideration the analysis capabilities on which these communities have come to rely. Continued support is needed for both the hardware and software resources that are presently deployed during the period of transition into newly developed processes and methodologies.

Strategy	Design with productization in mind.
Partners	All of ASC
Priority	Essential
Difficulty	Easy
Newly developed software and systems should be designed from the earliest stages to ultimately transition to a production role. By leveraging software architectures and software design strategies that simplify (or maximize the efficiency of) testing, troubleshooting, and long-term maintenance, developers ensure that newly developed capabilities will be reliable and economically maintainable throughout their service life.	

Strategy	Maintain tight integration with IC program element.
Partners	ASC Integrated Codes
Priority	Essential
Difficulty	Easy
Ultimately, visualization and data analysis resources must be coupled with the simulation resources provided by the IC subprogram, providing an integrated capability for the user community. Establishing a collaborative relationship with IC and designing for an integrated capability from the earliest stages will ensure minimal expense to the ASC Program in delivering these capabilities. It also will minimize the longer term troubleshooting and maintenance burden.	

Strategy	Collaborate with external developers and vendors.
Partners	All of ASC and industry
Priority	Important
Difficulty	Medium
Laboratory staff is generally tailored for R&D activities and ill suited for production software support and maintenance. Leveraging contractual support obligations through external vendors will ensure greater economy and flexibility in acquiring the maintenance, support, modifications, and documentation typically required of a production code. Contracting with skilled external software engineering firms may allow for accelerated development while ensuring a path toward productization.	

Strategy	Migrate existing hardware resources into long-term infrastructure maintenance and renewal programs.
Partners	Acquisition teams, FOUS
Priority	Important
Difficulty	Easy
The advanced visualization facilities and other hardware resources born out of prior ASC R&D efforts have generally become a core component of the overall HPC capability delivered by the program. With continuing advances in commodity technology, these resources are becoming more reliable, more capable, and more readily available as off-the-shelf solutions. This, along with a shift in research focus as the HPC community moves toward exascale, has driven ASC research in this area to wane considerably. Moving forward, these resources should be treated as core HPC infrastructure, allowing for continued upgrades, scaling, renewal, or decommissioning as required by the needs of the user community.	

Strategy	Ensure evolutionary transition to new usage models.
Partners	ATDM
Priority	Essential
Difficulty	Easy
Revolutionary changes in workflow, analysis procedures, and software interfaces will increase the risk of poor adoption by the user community. However, continued reliance on antiquated analysis processes will degrade workflow efficiency and system utilization of pre-exascale systems. Newly developed data analysis capabilities should be architected or strategically introduced to allow for tractable and timely adjustment to users' workflows and expectations. Well-crafted training and documentation materials and programs should also be considered before deployment of any new capability or methodology.	

CONCERN: Appropriate Investment in Hardware Is Crucial to the Success of Exascale Data Analysis and Visualization

The existing tri-lab strategy of a combination of shared resources at the platform and local resources at individual sites should be continued. Shared sites can optimize for a community of tri-lab users, while individual sites can provide a flexible environment for local users that satisfies a need for on demand, interactive resources. This strategy will make the most effective use of the investment in petascale computing by providing a range of resources that will optimize compute time as well as customer effectiveness.

Strategy	Include consideration of visualization and analysis hardware as part of the platform procurement.
Partners	Platform acquisition teams
Priority	Essential
Difficulty	Easy
Data analyst and visualization specialists strongly recommend that visualization and analysis hardware resources be consistent with platform costs. Visualization and analysis resources are often tacked on as an afterthought, especially when budgeting. Exascale demands that integrated solutions be designed and delivered in concert, so that results can be analyzed. Note that this includes not only the platform-specific visualization and analysis resource and the rendering resource, but also the storage, I/O system, as well as the local area networks (LANs) and wide area networks (WANs) connecting the compute platforms to any visualization clusters.	

Strategy	Include dedicated visualization resource near the compute platform.
Partners	FOUS
Priority	Important
Difficulty	Medium
Set up an appropriately sized conventional visualization cluster sharing a parallel file system with the compute platform. This provides a flexible platform that can serve many use cases, run standard software, and can be upgraded at a rate different than that of the exascale platform.	

Strategy	Include appropriate local rendering and capability visualization and analysis resources in site planning; coordinate as needed across the sites.
Partners	Platform acquisition teams, ASC management
Priority	Highly Important
Difficulty	Medium
Data analyst and visualization specialists strongly recommend appropriately sized local hardware resources, including the rendering and capacity visualization clusters and desktops, the networking and I/O resources that enable remote access of shared tri-lab computing resources, appropriate connectivity to the analysts desktops, and the visualization facilities used to view the results of visualization and analysis. These should be coordinated with other local resources and customers. This addresses the user's need for a highly available, dedicated resource for data analysis and visualization with shared access to the data, and can be used for local rendering as well as local visualization of capacity runs. Note that cost will depend upon specific site needs. Due to the individual nature of the work performed by user communities at each lab, these facilities should be provided per local requirements.	

Strategy	Upgrade performance of local hardware over time, consistent with platform capabilities and applications requirements.
Partners	ASC strategic planners
Priority	Highly Important
Difficulty	Medium
Hardware upgrades are particularly critical in any area affecting bandwidth into the processors, such as the secure WAN, I/O or Peripheral Component Interconnect Express (PCIe) into the graphics processors. Hardware components such as PCIe interfaces, which will double in speed twice over the next five years, may provide easy solutions to bandwidth problems encountered as users manipulate more exascale data. Graphics cards have been doubling in speed every year and may provide the ability to render faster and provide other means of analyzing user data. These solutions could potentially provide improved data analysis and visualization services to end users and should be considered where appropriate for customers. An analogous situation on a platform would be the upgrade to faster processors to provide more compute power.	

Cross-Cutting Challenges

The type of computer architectures that will support ASC simulation and modeling capabilities at petascale are undergoing a revolutionary transition. The next generation of processor architectures will continue to increase performance but will do so in a disruptive way that will require changes to existing visualization applications at the algorithmic and coding levels, just as changes will be required for ASC simulation codes. These architectural changes are introducing a number of cross-cutting research challenges:

- **Software stack**. The push toward exascale will necessitate the adoption of novel architectures and the abandoning of antiquated standards. Compatibility of the full software stacks associated with ASC visualization suites may become a challenge.
- **Resource contention**. The emergence of NVRAM in the system creates potential conflicts in resource utilization. How does the ASC Program simultaneously exploit NVRAM for application (user-driven), analysis (library-driven), and resilience and I/O (system-driven) use cases?
- **Data management and provenance**. The proliferation of complex workflows will strain current ad hoc data-management methods. Leveraging modern technologies to capture provenance and aid users in managing simulation setup, output, and analysis/visualization products is highly recommended.
- **Resilience**. There will be a significant increase in components on future architectures, without a significant decrease in the mean time between failures for each one. As such, the community is expecting a significant increase in fail-stop errors, transient errors, and silent data corruption at exascale. The visualization and analysis communities will need to embrace the new resilience strategies being developed to mitigate the increase in faults on future systems.

CONCERN: Development and Adoption of Capabilities that Span the Software Stack Require Tight Interdisciplinary Coordination, Introducing Significant Social Challenges in Addition to Technical Challenges

Strategy	Continue investment in interdisciplinary efforts and co-design.
Partners	Industry, universities, DOE laboratories
Priority	Highly Important
Difficulty	Medium
Facilitate communication across all aspects of the software stack. Invest in projects and workshops that encourage interdisciplinary communication and coordination to iteratively co-design all aspects of the exascale ecosystem.	

Strategy	Define and impose requirements on networking, file systems, and storage areas to support remote access and to support access to the file system and other local compute resources.
Partners	FOUS
Priority	Highly Important
Difficulty	Medium
Performance and capabilities of exascale networking and storage services will directly impact visualization and analysis capabilities. In situ and in-transit capabilities mentioned in previous sections require performance-portable networking and storage capabilities beyond what exists in today's systems. Development and deployment of efficient and reliable application workflows require capabilities for provenance capture and search that do not currently exist in today's storage systems. Finally, even in an exascale environment, remote users will still require reliable and secure access to data generated on HPC platforms. Analysis and visualization are inherently limited by the capabilities of the networking and storage system. The analysis community must work closely with the networking and I/O teams to drive requirements for both performance and advanced capabilities to directly support the needs of analysis and visualization on extreme-scale platforms.	

CONCERN: Commodity Hardware Technology Continuously Advances and May Provide Opportunities for Transformational Capability in Visualization and Analysis

There are several technologies emerging or expected to improve greatly over the next ten years. Among these trends are the improvement in GPU performance, the collocation of the GPU and CPU onto a single die on some chipsets, and improvements in the cell and similar processors. All have the potential to improve visualization and rendering performance on ASC problems.

Over the last five years, GPUs have been improving in processing power at a rate of 2x to 2.5x per year. This means there will be impressive compute power available at the end of the visualization pipeline, disproportionate to the rest of the system. Finding ways to exploit this power to directly impact ASC users may provide breakthrough capabilities that enable better understanding of ASC data. The collocation of CPU and GPU on the same die presents an opportunity. The bandwidth will be much greater coming from the CPU into the GPU, thus addressing the PCIe/GPU imbalance. Streaming visualization/rendering techniques could be used to exploit this. A third area that must be kept in view is cell and similar accelerator technology as they advance. Other areas of interest may present themselves in later years.

Strategy	Continually stay abreast of commodity hardware and software developments and investigate ways in which software and algorithms designed for such hardware could produce benefits that could be applied to ASC exascale needs.
Partners	ATDM
Priority	Essential
Difficulty	Hard
Because these developments have the potential to provide solutions to the needs of ASC's large data customers, resources must be devoted to staying current with advances. By collaborating with colleagues, staying informed, and experimenting as necessary, there is an opportunity to develop software that takes advantage of specific capabilities of leading-edge hardware in novel ways to solve important problems for ASC users. This should be a user-centered approach in which knowledge of the state-of-the-art combined with intimate understanding of user requirements will allow informed decisions about applying these technologies.	

Input/Output, File Systems, and Storage

Introduction and Background

In the next five years, both leadership-class storage systems and smaller capacity storage systems will be deployed for use by scientific applications throughout the NNSA complex. Further, the research, planning, and design phases for follow-on leadership-class storage systems will be largely completed. In 2015, Los Alamos National Laboratory (LANL) will deploy a storage system to support the Trinity supercomputer. This storage system will be the first leadership-class system within DOE to leverage both a parallel file system and a burst buffer composed of NAND flash solid-state memory. This first experimentation with adding NVM to the memory hierarchy in the storage system will result in an 82-PB file system supporting 1.45 TB/s of bandwidth and a 3.7-PB flash tier providing 3.3 TB/s of bandwidth. In 2017 and 2018, LLNL will deploy a storage system for the Sierra machine, providing a 120-PB, 1-TB/s parallel file system as well as a fast non-volatile, node-local memory tier. In the path from Trinity to Sierra, the ASC Program sees the parallel file system's role diminishing to focus on meeting pre-exascale data capacity requirements while the high-bandwidth requirements will be largely met with a fast non-volatile storage level.

The I/O bottleneck—the performance disparity between the high rates at which applications want to write data into the network versus the much lower rate at which storage systems can store data to file systems—continues to limit the rate at which scientific applications make progress. As machine failure rates and machine scale continue to increase in tandem, reliable storage systems are the only mechanism available for scientific applications to ensure they make progress and validate the generated results. The tightly coupled nature of scientific applications continues to impose a unique set of demands and considerations in the design of leadership-class storage systems. For example, unlike popular commercial and data-analysis applications, scientific applications routinely generate *single-source* datasets (possibly even single files) in the range of multiple petabytes. Additionally, the NNSA has historically been a leader in requirements for data retention. One result of the requirements to combine large volumes of single-source data with long-term data retention is that leadership-class storage systems will need to be tightly coupled throughout the storage hierarchy; that is, both Trinity and Sierra will have compute facilities that require the careful scheduling of burst buffer and parallel file systems resources and attention to incorporating in situ analysis and other techniques to reduce the intermediate data written to storage while still ensuring sufficient data to generate desired scientific insights.

Storage system design has entered an age of intense interest. Large commercial entities such as Amazon, Google, and Microsoft operate huge, geographically dispersed storage systems. Although their workloads are dominated by loosely coupled applications storing and retrieving multiple data sources simultaneously, their economies of scale are compelling and challenge us to leverage their technologies and methodologies where appropriate. Further, it is in the Labs' best interest to continue monitoring the evolution of commercial storage system design to find opportunities for convergence where possible.

Finally, the tight integration between each level within the storage hierarchy requires broadening of the scope of storage system research and design. It is now critical to begin architecting an extended memory hierarchy that includes storage systems, and that tightly integrates with resource managers, job schedulers, and data management throughout the application workflow.

Common Storage Systems and Input/Output Concerns and Strategies

A number of concerns and strategies were common across extended memory hierarchy, file system, and archive focus areas.

CONCERN: Ability of Laboratories to Address Exascale I/O Challenges on Their Own

The number and type of challenges presented by exascale I/O exceed the ability of the ASC Program to provide solutions in isolation. It is vital to leverage and collaborate with academia and entities funding technology solutions.

Strategy	Leverage collaborations and research investments.
Partners	Office of Science, PSAAP, Design/Fast/PathForward, Storage Systems and I/O (SSIO)
Priority	Highly Important
Difficulty	Medium
There are a wide variety of DOE collaborations and research investments that should be fully supported and leveraged, including DOE DesignForward, FastForward and PathForward efforts; ASC Program university investments; the DOE SSIO initiative; the ASC PSAAP II; the National Security Applications Co-Design Project; and Advanced Scientific Computing Research (ASCR) X-Stack projects. In addition, single (for example, Defense Advanced Research Projects Agency (DARPA)) and multi-agency collaborations should be leveraged and standards bodies should be engaged. Whenever possible, ASC Program I/O projects will take advantage of the open source software community and projects.	

CONCERN: Requirement for End-to-End Integration of Storage Resources with Scheduling Systems

Today’s HPC scheduling solutions focus almost exclusively on coordinating and managing compute resources. Successful exascale workflow systems will require that all aspects of computing, network, and storage resources work together and are scheduled together. In particular, data being stored, staged, and drained from NVM will require proper scheduling, charging, and allocation.

Strategy	Integrate storage system and I/O resources into HPC schedulers.
Partners	ASC Program, ATS vendors
Priority	Highly Important
Difficulty	Hard
The integration of burst buffers into storage hierarchies will be the tipping point requiring schedulers to integrate the management of storage bandwidth and capacity into coherent workflow scheduling. Efficient staging and asynchronous draining of burst buffers, file systems, and archives will need to be incorporated into system- and center-wide allocation, scheduling and charging infrastructure in an efficient and secure manner. Bi-directional I/O will need to be co-scheduled as a peer with compute cycles. The tri-labs will work internally and with vendors to integrate storage resources into HPC scheduling systems.	

Strategy	Investigate, design, and deploy hierarchical storage solutions.
Partners	CSSE
Priority	Highly Important
Difficulty	Medium
Integrating new storage levels such as burst buffers and long-term storage enables a configuration with a namespace spanning across the storage hierarchy. Utilizing this configuration, the namespace for data on the burst buffer can be established on the parallel file system or long-term storage solution while the data takes up residence on the burst buffer and later is moved through the storage hierarchy. This can enable better provenance of data as well as flexibility in data locality. Systems configured with this type of namespace spanning are available via commercial applications; development is needed to make a similar solution available for ASC systems.	

Memory Hierarchy

The limited realized I/O bandwidth of current platforms is restricting Parallel File System (PFS) use for both resilience and intermediate data storage. An emerging solution of offering non-volatile memory devices paves a way to offload data from the compute node random access memory (RAM) prior to moving to the PFS or paves a way to avoid the PFS entirely. This new NVM layer is currently seen as a cache or buffer for the PFS. While this is useful, it is a limited view of what may be the best way to take advantage of NVM resources to reduce pressure on the PFS as well as to enhance Integrated Application Workflow (IAW) throughput. Several concerns have been identified related to NVM and are discussed below.

CONCERN: Successfully Leveraging Burst Buffers

Efficient use of burst buffers is required to reduce file system bandwidth and capacity requirements by an order of magnitude in the next three years. The use of NVM for burst buffers is in its infancy and will require significant investment to deliver on this requirement.

Strategy	Implement Trinity and Sierra burst buffer.
Partners	CSSE, FOUS
Priority	Essential
Difficulty	Medium
Trinity will be the first ASC advanced technology system with an NVM layer. It will reside off-node. The Sierra machine will follow with an on-node NVM layer. DOE laboratory and vendor partners will work closely to integrate this new memory layer into advanced technology system architectures and ASC application workflows. Particular areas of emphasis will include application resilience strategies such as checkpoint/restart, data security, naming, scheduler integration, wear reduction, use quotas, data refactoring, N-to-1 and N-to-M file access solutions, and the standardization of NVM APIs.	

CONCERN: Burst Buffers Do Not Solve the PFS Input/Output Bandwidth Problem

NVM resources in the form of a burst buffer or other in-compute area storage partially address the burst I/O bandwidth mismatch between the PFS and the compute area. The remaining performance gap manifests when significant I/O bandwidth is required to drain NVM resources to the PFS.

Strategy	Design and construct integrated analysis code suites.
Partners	Integrated Codes (IC), CSSE
Priority	Important
Difficulty	Medium
Evolve from using the PFS for intermediate data storage to incorporating an IAWs/in situ analysis configuration to reduce pressure on the I/O bandwidth by keeping unneeded intermediate data from being written to the PFS using NVM or other in-compute area resources and techniques.	

CONCERN: Sharing Burst Buffers with Integrated Application Workflow and Parallel File System Needs Investigation

IAWs require high-bandwidth, low-latency data storage access from compute resources. Resources for storing this data outside the limited compute resources and away from the limited PFS performance would enable this richer scientific environment.

Strategy	Expand PFS into NVM.
Partners	CSSE
Priority	Important
Difficulty	Medium
Treat NVM data storage resources as an extension of the PFS and extend metadata from the PFS into the NVM layer. This will require additional research into data migration partially informed by hierarchical storage systems but with data that should never migrate or be discarded unilaterally by the metadata service as demand for the storage space arose.	

Strategy	Leverage key-value interfaces for burst buffer.
Partners	ASC, HPC vendors
Priority	Important
Difficulty	Easy
It is not clear that a POSIX-like interface or a block interface is ideal for intermediate data. Instead, consider offering Key-Value store or other metadata interface for NVM resources, keeping them outside of the I/O path.	

CONCERN: Planned Burst Buffer Focus on PFS I/O Bandwidth Absorption Potentially Prevents IAW Research

The current strategy for NVM is primarily to use it to store checkpoint data for resilience. By using NVM for example, for data staging between IAW components, application codes can avoid PFS I/O bandwidth limitations and enhance the science. The co-design teams need to consider how to address a diverse set of uses for NVM resources to get the best benefit from the technology on these and future platforms. These devices are purchased with a dual use in mind, and this dual use must be enforced in policy and support.

Strategy	Ensure shared priority for at least a portion of burst buffer resources for other uses.
Partners	FOUS
Priority	Important
Difficulty	Easy
Reserving a portion of burst buffer resources for non-PFS caching purposes. Application and system uses such as analytics or persistent data sets are simple approaches to guarantee the broader research agenda for this hardware.	

CONCERN: NVM Limited Write Endurance

Since the NVM resources likely for the next five years may have limited write endurance, this characteristic must be considered in any management scheme.

Strategy	Include write count as part of cost.
Partners	ASC Program, HPC vendors
Priority	Important
Difficulty	Easy
Allocate NVM usage based on device lifetime fraction as another resource.	

Strategy	Allocate number of writes as part of compute award.
Partners	ASC Program, HPC vendors
Priority	Important
Difficulty	Easy
Degrade availability and use for users that disproportionately wear write-limited devices. This would require support from scheduling services.	

File Systems and Structured Storage Systems

For the pre-exascale procurement cycles, parallel file systems are expected to be a significant expenditure, but these expenditures should diminish over time. The national laboratories all recognize that a reliable parallel file system is still necessary for applications to make progress; however, extended memory hierarchies (deployed as burst buffers) and disk-based, long-term storage are expected to remove the emphasis on deploying traditional POSIX-compliant parallel file systems. However, the need for structured storage systems will not disappear, and the desire exists to begin deploying storage systems based on objects and key-value hashing to better leverage commercial industry trends.

CONCERN: Storage System Reliability at Scale

Parallel file systems continue to exhibit brittle behavior at scale. Network failures, media performance degradation (including media failures), and complex coherence protocols all contribute to a general lack of reliability-at-scale for parallel file systems. Unlike compute systems and extended memory hierarchies, storage systems are required to provide a high level of data reliability and thus must provide higher levels of availability and reliability. Further, these advancements in availability and reliability must occur at system scales exceeding those currently deployed.

Strategy	Deploy alternatives to traditional RAID.
Partners	FOUS, storage vendors
Priority	Highly Important
Difficulty	Easy
Pre-exascale deployments are expected to decouple the data reliability algorithms in use and the time required to rebuild data in the face of a media failure. As hard drives have increased in size, so have the rebuild times required by traditional RAID algorithms. Future file system procurements will leverage software-based parity algorithms and pool-based replica selection. Additionally, future Lustre file systems will also deploy GridRAID, a software-based RAID controller that succeeds in decoupling data protection algorithms from failure rebuild efforts. Data center-wide file systems, such as Ceph, are now deploying with software-based and erasure-coding-based data protection, and efforts to use erasure coding throughout the reliable storage tiers are critical in reducing the costs associated with reliably storing data quantities.	

Strategy	Leverage robust networking middleware.
Partners	CSSE, FOUS
Priority	Important
Difficulty	Hard
The capability of system software layers to adapt to network faults is of particular concern for file systems. Current software continues to exhibit heavy reliance on connection-based network protocols, which have not proven to be fault tolerant at large scale. Recovery methods are heavy network users and have been known to cause disruption of other services sharing the I/O network. Deploying networking middleware within the storage system is a critical piece in building file systems and structured storage systems that reliably scale and are resilient to frequent failures within the compute and storage partitions.	

CONCERN: Scalable Metadata Performance

Scaling of metadata operations in both file systems and high-performance storage system (HPSS) has not been solved. While some engineering solutions have been put forth for scaling, some metadata operations—including name space division, directory hashing, and directory splitting—there are no solutions that address this area fundamentally. Metadata will be used by both the system and by applications, and given the number of processing elements of future supercomputers, this is one of the top problems for ASC. Additionally, extensible metadata and alternatives to tree-based metadata organization and access need to be explored to help with management of the billions of files ASC expects to manage in the pre-exascale timeframe.

Strategy	Research, develop, and deploy distributed namespaces.
Partners	CSSE, OpenSFS, Intel or other Lustre support organizations, IBM
Priority	Important
Difficulty	Hard
Distributed namespace performance has continued to lag behind the metadata performance desired by many users. Although Global Parallel File System (GPFS) and Lustre both have distributed namespace implementations, neither has yet eliminated the metadata bottleneck. Continued investments in improving the performance of the available distributed namespace implementations are important to continue slowly improving overall metadata performance.	

Strategy	Accelerate small, unaligned data transfers and metadata operations.
Partners	CSSE, FOUS, HPC vendors
Priority	Important
Difficulty	Medium
Metadata accesses are slow due the small, unaligned nature of the media access, an access pattern with which spinning disk has traditionally performed poorly. Emerging non-volatile memory types promise much higher random read access and somewhat higher random write access. Additionally, it is necessary to architect low-latency networks into structured storage system software to ensure complex metadata operations are not limited by network startup costs.	

Strategy	Investigate alternative metadata organizations.
Partners	CSSE, industry, standards organizations
Priority	Highly Important
Difficulty	Hard
The traditional inode-based storage organization for metadata has not seen significant innovation in decades. However, recent interest in log-based metadata organizations demonstrates an opportunity for HPC research efforts to dovetail with cloud-scale research. Efforts such as LevelDB and BatchFS are examining how to reorganize and optimize the storage of metadata into a write efficient format. Further research efforts into novel metadata layout are an important strategy for improving metadata performance.	

CONCERN: I/O Bandwidth Scaling

Parallel file system performance has historically relied on increasing the number of paths from the compute partition to the storage partition to maintain acceptable client/server performance. Providing uniform bandwidth to each storage cabinet becomes prohibitive as the storage systems grow to be composed of more than tens of thousands of disks. Rather, the program must leverage the trends within current leadership-system network architectures that provide high-bandwidth cliques, interconnected with much lower bandwidth links. A basic shift in architecture to hierarchical storage paths is inevitable and necessary.

Strategy	Eliminate POSIX bottlenecks.
Partners	CSSE
Priority	Highly Important
Difficulty	Hard
The strategy of an intermediate fast storage layer (burst buffer) is in initial deployment. Several vendors have begun to offer solutions in this area; however, the software needed to implement a storage hierarchy is not well tested. Preliminary research is examining fully hierarchical storage software stacks capable of storing data to user-specified safety and retention levels. Continued research into HPC-focused object storage tiers, including ongoing efforts such as Triton and Sirocco, are useful in that they explore the convergence of more traditional bulk-synchronous parallel HPC and other workloads.	

Strategy	Leverage advancements in checkpoint algorithms.
Partners	IC, CSSE
Priority	Important
Difficulty	Medium
Traditionally, the storage of checkpoint state is isolated from all earlier checkpoints. As the system memory sizes for systems such as Trinity and Sierra grow, it is necessary to explore techniques for compressing the size of the checkpoint data stored. Native file system compression is one technique that may be implemented at multiple levels within the file system software stack. Additionally, the exploration of techniques for storing the differences, or deltas, between sequential checkpoints rather than persisting the entire checkpoint is critical to lessening the I/O bottlenecks existing at all levels of the HPC complex. Opportunities to compress checkpoints exist between the compute partitions and burst buffers, between the burst buffer and the backing store, and between all tiers in the persistent storage hierarchy.	

Archive And Long-Term Storage

For twenty years, HPSS has provided high-performance archive service for ASC environments. Today the ability to generate information and the need for rapid access to this data in ASC environments are adding new requirements and leading centers to search for strategies for rapid and repeated access to long-term data storage while maintaining archive-level data integrity and security.

CONCERN: Time to First Byte of Long-Term and Archive Storage

File read demand is growing, as is the need to quickly retrieve data from long-term storage. Time to first byte on tape devices is slow, leading to a requirement for the judicious cost-effective insertion of faster access technologies in front of tape archives.

Strategy	Develop intermediate storage tiers—Campaign Storage.
Partners	FOUS
Priority	Highly Important
Difficulty	Medium
Campaign storage refers to a layer of quota-controlled storage between traditional scratch file systems and tape archives, allowing for rapid access to data with long-term residency (not subject to purge). It is based on the use of an erasure code and a commodity disk that provide high data integrity at a very low cost for the available performance. The use of commodity disk provides for a 66-percent reduction in cost per deployed unit of storage compared with enterprise disk used in parallel file systems, and a 50-percent reduction in enterprise disk configurations typically used in HPSS disk caches. ASC Program centers will deploy/expand this storage to allow for rapid access and search. Initial deployments are being implemented and future work is focused around utilization of object storage, data security, powering down disk, snapshotting, and versioning.	

Strategy	Expand HPSS disk caches.
Partners	FOUS
Priority	Highly Important
Difficulty	Easy
Expanding the capacity of HPSS disk caches will achieve many "time to first byte" goals while leveraging existing, highly reliable and available HPSS infrastructure. Expanding disk caches to improve cache hit ratios and reduce tape stage activity will enable rapid data access and a reduction in the number of archive tape drives required, and the solution will integrate seamlessly with existing HPSS namespaces.	

Strategy	Add tape ordered recall and mount awareness in HPSS.
Partners	CSSE (HPSS development teams)
Priority	Important
Difficulty	Easy
Physically mounting and positioning tapes accounts for a large portion of archive access times. Enterprise tape drives have added APIs that return optimal access plans for data access on tape. The ability to leverage these APIs will be added to HPSS along with added intelligence in tape drive/type selection to optimize data access times.	

CONCERN: Scalable HPSS Metadata Performance

As the number of archive files grows, HPSS metadata services are a strain to maintain performance. Initial scale-up modifications have been developed that improve HPSS metadata performance, but scale-out improvements remain in development.

Strategy	Develop multiple metadata servers in HPSS.
Partners	CSSE, HPSS developers
Priority	Highly Important
Difficulty	Medium
The work planned for HPSS metadata/core server scaling, targeted for HPSS Release 8, will enhance the ability ASC Program archives to scale in support of exascale compute resources.	

CONCERN: Security and Integrity of Archived Data

Data stored in the 1960s is still retained in ASC archives. Data destined to be stored for multiple decades must be protected from corruption, loss, and security breach.

Strategy	Encrypt HPSS interfaces and data at rest.
Partners	CSSE, HPSS developers
Priority	Important
Difficulty	Medium
There are currently no encrypted HPSS data transfer interfaces. Interfaces such as Secure File Transfer Protocol (SFTP) and Secure Copy (SCP) will be investigated for possible HPSS integration. Vendor solutions for encrypting data at rest exist but are varied and not standardized. Encrypted tape solutions will be surveyed and deployed as appropriate.	

Strategy	Archive scrubbing and automatic checksums.
Partners	CSSE, HPSS developers
Priority	Important
Difficulty	Medium
Over decades, archive data will be copied many times to new storage technologies. Aside from user-driven checksums, there is little guarantee that data is not corrupted or changed over time. Methods for automated checksums and integrity checking of archive data will be investigated and integrated as appropriate into existing archives.	

Networks and Interconnects

Introduction and Background

The network interconnect environment encompassed by the tri-lab ASC network is a tightly coupled confederation of LANL, Lawrence Livermore National Laboratory (LLNL), and Sandia National Laboratories (SNL) classified supercomputing facilities. The environment scales in many dimensions. Each lab network environment spans from the central computing rooms to the user's desktops. The network performance starts at 1 gigabit/s at the desktops and increases to the multi gigabyte/s interfaces in the core of the compute platforms. There are tens of thousands of interfaces in the compute core, hundreds of connections between large resources, thousands of desktops, and a small number of links in the WAN connections. Each of the local laboratories' networks is connected across secure 1,100 mile, 10/100 gigabit/s wide area links.

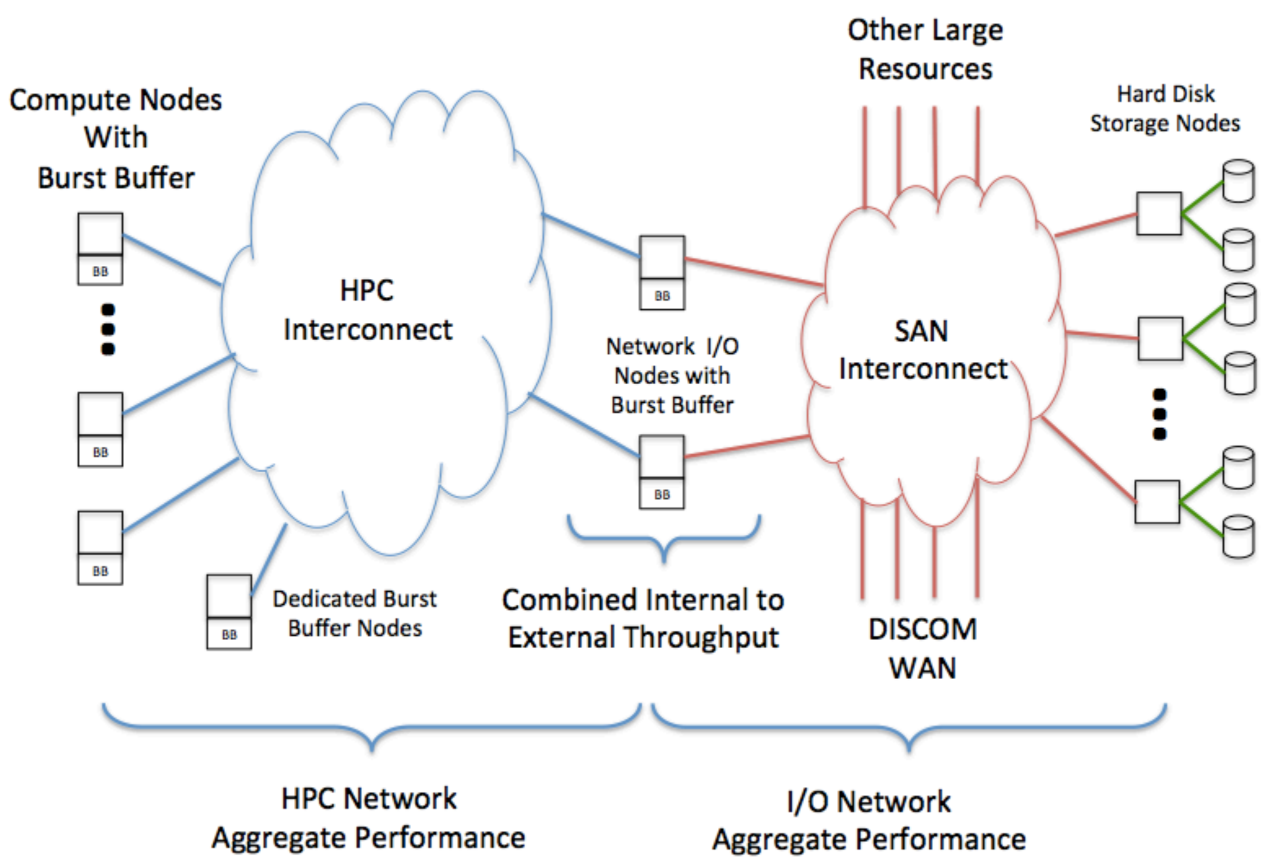

Figure 1. Combined ASC compute environment.

To effectively address this wide variety of scales, this section is divided into three subsections (WAN Interconnect, Storage Area Network (SAN) Interconnect, and HPC Interconnect). The HPC Interconnect subsection considers the extremely high-performance networks that interconnect the compute nodes inside the largest platforms. The SAN Interconnect subsection considers the network that interconnects the largest platforms with themselves, their resources such as storage, and the user's desktop. The WAN Interconnect subsection considers the network that interconnects the three laboratories in a seamless environment for sharing remote resources. The figure shows how these networks combine to provide the tri-lab ASC compute environment.

Wide Area Network Interconnect

The WAN interconnect between the ASC compute facilities consists of leased, private bandwidth and the associated equipment to securely interconnect the networks from each of the laboratories. The bandwidth currently consists of a 10/100-gigabit/s (Gbps) ring. Because the ASC mission is focused on classified products, the WAN links must be protected utilizing National Security Agency (NSA)-approved Type 1 encrypted devices. The following four technical challenges represent the consensus thinking of the tri-lab networking community; however, there are other challenges to designing a tri-lab network that are not technical in nature but will have an impact on planning and implementation. These challenges are the geographical disposition of pre-exascale computational resources and the evolving customer usage model. Consideration of these challenges will play a large role in solving the four technical challenges outlined below.

CONCERN: Slow Development of High-Speed Type I Internet Protocol (IP) Encryptors

Type I encryptor development has traditionally lagged the needs of the HPC community. As the pace of HPC development quickens, this lag continues to grow. The bulk of deployed Type I IP encryptors are still at 100 Mbps and 1 Gbps with 10 Gbps representing only a fraction of the units deployed. The 10-Gbps Type I encryptors were not available for delivery until September of FY07. The lag between 10-gigabit Ethernet availability versus 10-gigabit Ethernet encryptor availability was about five years. The 10-Gbps encryptors are currently deployed, replacing the parallel 1Gbps configuration with increased overall reliability. The needs of the HPC community are diverging from the broader market, and the encryptor vendors are not inclined to develop products focused on the small tri-lab community. The slow development of IP-based Type I encryptors is having a negative impact on the cost and efficiency of the current ASC WAN, and the impact will only worsen. A pre-exascale network environment can utilize 100-Gbps encryptors within five years as 40-Gbps NICs and 100-Gbps switch ports are currently commercially available. This challenge will be hard to solve, with the most likely successful approach being partnership with the larger community of the Department of Defense (DOD) and DOE to drive vendor development efforts.

Strategy	Perform faster development of high-speed type I IP encryptors.
Partners	Commercial companies, other government parties, tri-lab
Priority	Highly Important
Difficulty	Hard
The tri-lab community should take a more direct approach to obtaining the required encryptor technology. There are two levels of effort that the tri-lab community can pursue to further this goal. The first is to participate in requirements development and design reviews. The second is to encourage and support vendors entering this market by actively testing and using products that seek to improve the encryption performance. Because the ASC community has been very active in these two efforts, it has had some influence on encryption directions and products. In the past, ASC has used a third approach of directly contributing to early product availability by providing development funds. This third level of effort may be the only way that encryptors greater than 10 Gbps become available in the next five years because the rest of the community has not organized to push for faster capabilities. For example, ASC has already started discussing options with the current 10-Gbps encryptor vendor, L3 Communications Corp. They have fully developed and tested a new type of layer 2 encryptor based on extensions of the new MACSEC standard. They are waiting for a guaranteed market to develop that justifies the Type 1 certification costs. The tri-lab ASC community could provide significant motivation for L3 to complete the certification process.	

CONCERN: Low Reliability of Current Encrypted WAN Communications System

Type I encryptors are complicated devices in their own right and represent significant operational challenges to the tri-lab community. However, the community initially had to deploy them in parallel configurations to deliver the performance required, which only exacerbates the operational challenges and reliability of the network. Installing multiple parallel data paths was the solution adopted by the ASC WAN community to obtain higher bandwidth until 10-Gbps IP encryptors were available. Unfortunately, this configuration decreased the reliability of the services because a failure in any of the parallel units caused total link failure. The deployment of an architecture based on single 10-Gbps paths significantly improved the reliability of the ASC WAN. Returning to a parallel encryptor architecture due to limited encyrptor performance would be a step backwards in network reliability improvement.

Strategy	Increase reliability of encrypted WAN communications system.
Partners	Tri-labs, FOUS
Priority	Essential
Difficulty	Medium
To address the reliability issue involves taking a system-wide view of the entire ASC WAN environment and looking for a set of technologies that can be combined to provide significantly increased reliability. One of the proposed elements of a system solution is to apply existing network protocols in ways to ameliorate the problem of encryptor failure. In addition, limited network hardware redundancy can be installed to eliminate certain single points of failure. The developed system solutions could be immediately applied to the existing ASC WAN, thus providing benefits to the entire HPC community. Indeed, to realize the full potential of these solutions, the deployment of the developed solution would have to be tri-lab wide.	

CONCERN: Maintaining Efficiency and Performance while Scaling Network Protocols

Encryptors are not the only technologies that have to be stretched to meet the needs of the tri-lab community. The Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite that is the basis of contemporary computer networking was not designed for the WAN envisioned for the pre-exascale environment. Again, parallelism is used to provide the performance required, and operational difficulties are increased and reliable performance is decreased. Recent network protocol developments appear to provide a sound basis for increasing performance in the high-bandwidth plus long-delay WAN environment. However, standard industry development is unlikely to provide the required capability because the anticipated pre-exascale WAN environment is well beyond standard industry deployments.

Strategy	Scale network protocols to maintain efficiency for pre-exascale.
Partners	Commercial companies, universities, other government parties, tri-lab
Priority	Essential
Difficulty	Medium
The basic data networking protocols underlying the ASC WAN were developed over 25 years ago for a vastly different environment than that of the ASC WAN. The operating characteristics of these protocols in the WAN environment limit performance; however, recent developments provide opportunities to improve the situation. First, a number of additional performance tuning algorithms have been added to the protocol implementations. These new algorithms have the potential to improve performance if they can be tuned for the ASC WAN environment. A tri-lab development effort to test these alternative technologies would yield immediate benefits to the ASC WAN and has the potential to deliver the performance needed for the coming pre-exascale network environment. TCP/IP congestion algorithm development has shifted the default towards the CUBIC algorithm, which is a derivative of BIC (Binary Increase Congestion). An improvement to RED scheduling algorithm for the network scheduler named CoDel has also been developed to address ease of configuration and buffer bloat. Both are in development to test these alternative technologies. It is expected that further development in performance tuning has the potential to improve performance as the ASC WAN links are scaled to 100 Gbps.	

CONCERN: Managing, Operating, and Monitoring New Bulk Bandwidth Technologies

Given the nature of the increasing performance demands of the pre-exascale computing environment, new or novel solutions for delivering the bandwidth required must be examined. In the past, the tri-lab community has benefited from the dedicated network bandwidth available. However, it would only be prudent to examine alternatives such as bandwidth-on-demand or shared bandwidth offerings as ways to deliver the bandwidth required by the pre-exascale computing environment. Bandwidth-on-demand has not been offered to the tri-lab community in response to any of the service requests sent to the commercial service providers. It will take a special effort, working with the providers, to explore this service as an option. Similarly, it will take a special effort to define the needs of the tri-lab ASC WAN community in terms that could be addressed by a shared network environment because quality of service (QoS) parameters would have to be defined.

Strategy	Develop techniques for new bulk bandwidth technologies.
Partners	Tri-labs, FOUS
Priority	Important
Difficulty	Medium
The cost of the existing network is driven by the requirement to field very high bandwidth network links at all times so that the bandwidth is available when needed. This capability has been achieved through the installation of dedicated, high-speed communications links. It is prudent to examine alternate bandwidth solutions such as bandwidth-on-demand or shared facilities. Such solutions will involve working with vendors and national test beds to develop these services and the associated management capabilities to improve reliability and performance. Shared bandwidth services would require the implementation of end-to-end QoS if the ASC WAN is to maintain its current high performance. Deploying end-to-end QoS will require close collaborative development between the tri-lab community and potential bandwidth providers. In a similar vein, it will take collaboration between the tri-lab community and the vendors to develop a bandwidth-on-demand service model. If these or other new services can be deployed in the ASC WAN, they could be used immediately and provide significant cost advantages.	

Storage Area Network Interconnect

The SAN interconnect consists of the very large network that connects compute platforms to parallel file systems, visualization platforms, pre- and post-processing machines, archival systems, other compute platforms, and the WAN interconnect to the other sites. Each laboratory has unique implementations of this network, but they share important common characteristics: there are hundreds to a few thousand ports, some form of parallel networking is required to build the scale required, and Ethernet and InfiniBand (IB) are the existing technologies because they scale in distance (fiber) and are commonly used on all platforms. Each lab is also experiencing the same difficulties in this environment.

CONCERN: Inefficiency of SAN Interface Cards and Protocols

The most common data traffic that impacts the SAN concerns the flow of application data between the parallel and network file systems located on the SAN, and the nodes on the HPC network. The emerging burst buffer technologies should reduce the data pressure between the HPC network and SAN. Gateway nodes are typically configured to move data between the HPC and SAN networks, thus doubling the amount of data these servers must move. Sustained network throughput drops significantly in this scenario. The availability of sufficient node bus bandwidth may also limit the sustained throughput capabilities of these gateway nodes. Without sufficient hardware bandwidth, the protocols cannot possibly achieve required performance. There are several mechanisms that have been proposed to raise the efficiency of data movement through the servers. Because most of these mechanisms impact the data-movement applications, care must be taken to ensure the system as a whole continues to provide all of the necessary functionality as more efficient mechanisms are developed and deployed.

Strategy	Develop technologies that maximize effective bandwidth of network interface cards (NICs).
Partners	Vendors, CSSE, open-source concerns
Priority	Important
Difficulty	Hard
There are two potential paths to improve the efficiency of data throughput in ASC servers: efficiency of TCP/IP processing and taking advantage of the Remote Direct Memory Access (RDMA) protocol. Because TCP/IP is the dominant protocol for the SAN and Ethernet technologies, the ASC Program needs to work with the Linux community to improve the efficiency of TCP/IP processing. There are a few different mechanisms that may have potential for improvements, including splice, large segment offload, user space TCP, and TCP offload engines. The ASC Program should continue to participate with industry to complete the work on these mechanisms and ensure that they will function in ASC environments. The second potential path to very high efficiencies is to modify the ASC environment to take full advantage of the RDMA protocol. The existing RDMA standard is an extremely efficient mode of data movement, but it does not work the same way as TCP/IP. This requires extensive modification of many of ASC's existing data movement applications. It is not clear that these modifications would be possible due to both technical and vendor proprietary issues. There are other data movement applications already optimized for the RDMA protocol. The effective use of RDMA also requires an investigation into the use of new dispersive/adaptive routing schemes and congestion control methods for RDMA resource networks. The ASC Program community must investigate its ability to utilize RDMA and provide all of the required functionality. This is even more important now that the RDMA protocol can be utilized with Ethernet technologies as demonstrated in past ASC collaborations. The ASC Program community must also work with the motherboard vendors to ensure there is sufficient bandwidth in the I/O busses. There are new bus technologies such as PCIe v4 forthcoming in the industry; those or other faster technologies must be deployed for the pre-exascale environment.	

CONCERN: Managing Storage Area Networks and Potential New Technologies

The size and network traffic on the SAN leads to challenges in SAN management and operation. Given the large count of switch ports, physical infrastructure, NICs, and hosts that will be connected to the network, the probability that there will be failures at any given time is extremely high. This is the same problem experienced in the compute platforms with failures in the nodes, interconnects, and disks. The SAN environment must be designed for resiliency in the presence of failures anywhere in the system. Automated tools for detecting, isolating, and notification of failures will be critical to providing a reasonable mean time to repair for the system. If new technologies and protocols are deployed, then new mechanisms, techniques, and testing products will be required.

Strategy	Develop techniques for storage area network interconnect management technologies.
Partners	IB community, tri-labs
Priority	Important
Difficulty	Medium
The combined numbers of devices that will exist in the pre-exascale resource interconnect will be a serious challenge for any operational tools and processes. Automated tools for detecting and diagnosing failures must be developed and deployed in such a large environment. There will definitely be new technologies in this environment that will require new tool development. This could even include hardware devices for IB links, dynamic routing monitoring, and debugging tools. The ASC Program will partner with vendors to ensure products are available that meet ASC requirements. Management of the SAN will require much more than just monitoring the health of individual links and components. The complexity of the parallel architectures that will be implemented on top of the physical network will be significant. The tri-lab community must develop tools that will closely monitor the combined parallel communications to ensure the system is performing as designed. The goal of system resiliency to failures in the resource interconnect is not solely a function of network components. Architectures and applications that continue to provide critical services, even when there are failed components, require system-wide integration of applications and hardware. The networking team must work with the application developers and vendors to develop fault-tolerance at the network level. This is a primary focus of the Networking and Information Technology Research and Development (NITRD) Program research agenda, and ASC should participate and utilize their efforts wherever possible.	

High Performance Computing Interconnect

For all but the most embarrassingly parallel applications, the ability of the HPC interconnect must be balanced with the rest of the platform architecture to ensure good performance at scale. The transition towards extreme-scale computing is being achieved through higher parallelism within a compute node and by increasing the total number of nodes within a system. Both trends drive the need to increase the performance of traditional high-speed interconnect metrics while creating new requirements that were not significant in the petascale era.

CONCERN: Increasing System Imbalance

Technology advances have had a positive impact on both latency and message rate; however, the relative bandwidth increases continue to trail the increases in processor performance and memory bandwidth. Additionally, the seemingly inevitable move to lighter weight processing cores will have a dramatic impact on the message rate requirements for some communication models (that is, potentially reduced performance for MPI matching). One key challenge is how to provide continued sufficient interconnect performance in light of these technology trends.

Strategy	Partner with industry to engage in cutting-edge research and development in the area of high performance interconnects.
Partners	Intel, Cray, IBM
Priority	Highly Important
Difficulty	Medium
The ASC Program has historically been successful in influencing and, in some cases, driving developments in high-speed networks (both hardware and software). Examples include the Cray XT3/4 network, IB, and Myrinet. To ensure future architectures scale, it will be essential that the ASC Program continues to work with vendors to push interconnect capabilities beyond what is required for the bulk of the HPC market. Industry will continue to push processor architectures, which will provide the necessary floating point operations per section, but high-speed interconnects to tie together tens of thousands of such processors in an efficient matter is something that the DOE must drive. As such, ASC should establish strategic development programs with industry partners that specifically target high-speed interconnects.	

Strategy	Develop performance modeling and analysis tools to judge the impact of changes to key architectural features.
Partners	Tri-labs, CSSE
Priority	Highly Important
Difficulty	Medium
Although ASC Program, industry, and academia have made great strides in modeling and analysis capabilities, in many ways it is, as yet, insufficient to address detailed network analysis, especially at large scales. The ASC Program needs to increase their level of investment in this area to better define requirements, metrics, and future acquisitions.	

CONCERN: System Scaling

The HPC interconnect must enable rather than inhibit the scalability of applications and system services that require high-performance communication. Design decisions at the interconnect level can have direct impact on the amount of non-network resources required to enable efficient communication at full scale. For example, flow control and message buffering strategies that are based on usage patterns and that are independent of system size or the number of communicating endpoints have been shown to significantly reduce memory usage and increase network efficiency. Some fundamental operations, such as translating virtual to physical endpoint addresses, may also be infeasible as the number of network endpoints continues to grow.

Strategy	Place higher emphasis on HPC network requirements in future acquisitions.
Partners	CSSE, system acquisition teams
Priority	Highly Important
Difficulty	Medium
Although the ASC Program has been successful in deploying platforms with very good scaling characteristics, it will become imperative to push the vendors even harder in future acquisitions. As an example, the bound-to-free ratios are decreasing rapidly due to advancing processor developments, and the external market is not requiring high-speed interconnects at the level required for tri-lab workloads. The ASC Program can do this by developing a detailed set of requirements for future machines and providing realistic application and micro-benchmarks to measure performance metrics in as close to a real-world scenario as possible.	

CONCERN: Managing Scaled Interconnect Networks and Potential New Technologies

With the scale of the network growing to tens or even hundreds of thousands of end-points, the need for management tools to analyze and debug network issues is a must. These tools must also support reading of registers and other performance counters in the system for performance and power analysis. They must also be robust and resilient to failure. Performance modeling of future HPC interconnect architectures will require sophisticated performance modeling tools to best understand the impact that design decisions will have on the tri-lab workloads. ASC has a strong history and track record in providing national leadership and must continue to do so. Although the issues are common, each laboratory's requirements may not be of the same magnitude for any given capability. Each lab must continue to work with particular vendors to fit their requirement in addition to those of the other labs.

Strategy	Develop management and performance analysis tools for compute interconnect.
Partners	Common Computing Environment (CCE) program
Priority	Important
Difficulty	Medium
The ASC Program needs to work with industry to develop a tool or a set of tools to aid management and performance analysis at the network level. Current tools are non-existent or insufficient. This will also require working with vendors at the silicon level to define performance counters that need to be integrated into next-generation NICs and switches.	

Next Steps

This report identifies areas of concern that need to be addressed now by the combined efforts of the tri-lab ATDM/CSSE/FOUS subprogram elements. Some future work is well understood and needs merely to be managed and completed, while other concerns expressed here will require extensive study, analysis, collaboration, and innovation. Business as usual will not suffice for transitioning the ASC Program to the next generation of platforms and future exascale systems.

Complexity of the user environment will grow. Efforts to simplify the environment are required to enable productivity for the ASC Program mission. Additionally, efforts to assist users in developing efficient workflow strategies, data management approaches, visualization, and analysis processes will be necessary, as will expanding the number and types of tools available for performance enhancement, debugging, and tuning of performance to particular system characteristics. Collaborations beyond the NNSA laboratories to include the SC labs, the PSAAP II, and even open source organizations will be necessary to advance the state of the practice in software at both the application and system levels to meet the challenges of exascale.

Several technical areas of concern crosscut the five technical working groups:

- The additional layer of data management imposed by burst buffers, which are designed to improve the data transfer rates into and out of main memory, will require more complex workflow and scripting.
- Deep and complex memory hierarchies with varied performance, capacity, power, and volatility will require more complex application and systems data management.
- Programming models will need to be adaptable to permit targeting different architectures during the early move to exascale as architectures are likely to be varied and diverse, as seen in the differences between Trinity and Sierra.
- Unprecedented parallelism both in-node and across compute nodes will require complete rewrites of application code as in the ASC ATDM subprogram but will also require completely new approaches to system software; incremental change will not be sufficient.

Sierra and Crossroads planning and procurement activities are well underway. Concurrently, the ASC Program is adjusting to Trinity, the current AT system—Trinity—and hints of the challenges ahead are already visible. The ATDM and CSSE subprograms must work closely with the FOUS subprogram to apply lessons learned from integration and initial operations of Trinity and Sierra to identify early opportunities for adapting administration and management processes and policies, along with applications, to systems of a scale and complexity never before seen or managed. The challenges will be many; but with sufficient effort and forward-looking evaluations of technology choices, the nuclear weapons program can be assured that the ASC Program systems and environments will be available to maintain and support the national nuclear weapons program's computational simulation needs.

Appendix A. Acknowledgements

Technical Leads and Authors:

Michael Lang (LANL)
John Noe (SNL)
Rebecca Springmeyer (LLNL)

Authors and Contributors:

Janine Bennett (SNL)
Jim Brandt (SNL)
Robert Clay (SNL)
Trent D'Hooge (LLNL)
Parks Fields (LANL)
Scott Futral (LLNL)
Mark Gary (LLNL)
Daryl Grunau (LANL)
Pam Hamilton (LLNL)
Scott Hemmert (SNL)
Richard Hu (SNL)
Kathleen Kelly (LANL)
Ruth Klundt (SNL)
Kyle Lamb (LANL)
Dan Laney (LLNL)
Matt Leininger (LLNL)
Jay Lofstead (SNL)
Cory Lueninghoener (LANL)
Dave Montoya (LANL)
Ron Oldfield (SNL)
John Patchett (LANL)
Mahesh Rajan (SNL)
Martin Schulz (LLNL)
Bradley Settlemyer (LANL)
Aron Warren (SNL)
Paul Weber (LANL)
Mary Zosel (LLNL)

Appendix B. Acronyms

AMT	Asynchronous Many Task
API	Application Programmer or Programming Interface
ASC	Advanced Simulation and Computing
ASCR	Office of Science's Advanced Scientific Computing Research
AT	Advanced Technology
ATDM	Advanced Technology Development and Mitigation
ATDM	Advanced Technology Development and Mitigation
AWE	Atomic Weapons Establishment
BSP	Bulk Synchronous Parallel
CCE	Common Computing Environment
CEA	Commissariat à l'Énergie Atomique (the French Commission for Atomic Energy)
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
CSSE	Computational Systems and Software Environment
CT	Commodity Technology
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
DOE	Department of Energy
DRAM	Dynamic Random Access Memory
DSW	Directed Stockpile Work
FOUS	Facility Operations and User Support
FT	Fault Tolerance
FY	Fiscal Year
GPFS	Global Parallel File System
GPU	Graphics Processing Unit

HIO	High Performance Input/Output
HPC	High Performance Computing
HPSS	High-Performance Storage System
I/O	Input/Output
IAW	Integrated Application Workflow
IB	InfiniBand
IC	Integrated Codes
LAN	Local Area Network
LANL	Los Alamos National Laboratory
LDRD	Laboratory Directed Research and Development
LFLR	Local Failure Local Recovery
LLNL	Lawrence Livermore National Laboratory
LWK	Lightweight Kernel
MPI	Message Passing Interface
NIC	Network Interface Card
NITRD	Networking and Information Technology Research and Development
NNSA	National Nuclear Security Agency
NRE	Non-Recurring Engineering
NSA	National Security Agency
NVRAM	Non-Volatile Random Access Memory
NVM	Non-Volatile Memory
OS	Operating System
OST	Object Storage Target
PCIe	Peripheral Component Interconnect Express
PEM	Physics and Engineering Models
PFS	Parallel File System
POSIX	Portable Operating System Interface
PSAPP	Predictive Science Academic Alliance Program
QMU	Quantification of Margins and Uncertainties
QoS	Quality of Service
R&D	Research and Development
RAM	Random Access Memory

RAS	Reliable, Available, and Secure
RDMA	Remote Direct Memory Access
RTS	Runtime System
SAN	System Area Network
SC	Office of Science
SCP	Secure Copy
SCR	Scalable Checkpoint Restart
SFTP	Secure File Transfer Protocol
SIMD	Single Instruction, Multiple Data
SNL	Sandia National Laboratories
SSIO	Storage Systems and Input/Output
TCP/IP	Transmission Control Protocol/Internet Protocol
UQ	Uncertainty Quantification
V&V	Verification and Validation
VACET	Visualization and Analytics Center for Enabling Technologies
WAN	Wide Area Network

Appendix C. List of Strategies

The following table is a compilation of all strategies listed in this document.

Strategy	**Priority**	**Difficulty**	**Technical Area***
Maintain active involvement in standardization bodies.	Highly Important	Medium	PET
Research new abstractions for the existing models.	Highly Important	Hard	PET
Establish closer ties with programming model research efforts and increase early evaluation efforts.	Highly Important	Medium	PET
Support development of best practices and ultimately standardization.	Highly Important	Hard	PET
Increase design studies, in close collaboration between system, library, and application developers.	Important	Medium	PET
Develop and test new application programming interfaces for memory management.	Important	Medium	PET
Increase design studies, in close collaboration between system, library, and application developers.	Highly Important	Medium	PET
Develop new intra-node data locality and movement abstractions.	Important	Medium	PET
Develop a new technique for inter-node data staging that is also applicable to composite workflows.	Important	Medium	PET
Survey existing interfaces.	Highly Important	Medium	PET
Ensure tighter integration of introspection into the software stack.	Highly Important	Medium	PET
Standardize introspection interfaces.	Highly Important	Medium	PET
Provide introspection into job-level data through the resource managers.	Highly Important	Medium	PET
Integrate and provide access to machine-room level sensors.	Highly Important	Medium	PET
Develop tool-, system-, and application-agnostic tool	Highly	Medium	PET

Strategy	**Priority**	**Difficulty**	**Technical Area***
infrastructure.	Important		
Integrate tool infrastructures across components and make them available as a shared system service.	Highly Important	Hard	PET
Develop memory usage tools.	Highly Important	Medium	PET
Develop tools to better understand node-local and job-wide data transfers.	Important	Medium	PET
Develop tools and new metrics applying to task-based systems.	Important	Medium	PET
Develop tools to support porting to new programming models.	Highly Important	Hard	PET
Enhance current existing hardware and vendor-agnostic tools (possibly by integrating vendor-specific approaches).	Highly Important	Hard	PET
Develop tools to model and estimate benefits from porting to new architectures.	Important	Hard	PET
Extend existing debugging support for next-generation architectures and runtime systems.	Highly Important	Hard	PET
Enhance current existing hardware and vendor-agnostic tools (possibly by integrating vendor-specific approaches).	Highly Important	Medium	PET
Play an active role in the definition of new memory interfaces.	Important	Medium	PET
Play an active role in the inclusion of fault-tolerance (FT) techniques into runtimes and programming standards.	Highly Important	Hard	PET
Expand efforts in efficient checkpointing and integrate into overall software stack.	Highly Important	Medium	PET
Increase power instrumentation and access to power (and thermal) data in future machines.	Important	Medium	PET
Provide new generation of tools that enable the correlation of power, power caps, and performance.	Highly Important	Medium	PET
Develop high-level abstractions to hide the impact of power-aware programming.	Highly Important	Medium	PET
Develop power management system and policy.	Highly Important	Medium	PET
Develop performance models for applications under power caps.	Highly Important	Medium	PET
Expand efforts in the development of power-aware runtime scheduling and integration into other runtime efforts.	Important	Medium	PET

Strategy	Priority	Difficulty	Technical Area*
Co-design system management tools using tri-lab experience.	Highly Important	Medium	SMM
Utilize industry standard best practices for configuration management.	Highly Important	Medium	SMM
Provide a fast and reliable interface to all system components and implement mechanisms for recording communication failures.	Essential	Hard	SMM
Vendor must support rolling system updates.	Highly Important	Medium	SMM
Provide a hierarchy of service nodes.	Essential	Medium	SMM
Provide a fast and reliable interface to all components of system and implement mechanism for recording communication failures.	Essential	Hard	SMM
Provide capabilities for isolating failed hardware components.	Essential	Medium	SMM
Provide a scalable software management solution.	Highly Important	Medium	SMM
Develop scalable automated tools to quantify behavioral characteristics of applications.	Highly Important	Hard	SMM
Develop scalable monitoring and automated analysis tools to detect known degradation/failure modes.	Highly Important	Hard	SMM
Develop scalable automated tools to quantify behavioral characteristics of applications.	Highly Important	Hard	SMM
Utilize behavioral characteristics of the application mix during procurement process.	Highly Important	Medium	SMM
Develop scalable monitoring/analysis tools to identify inefficient use of resources.	Highly Important	Hard	SMM
Develop scalable component power monitoring and profiling tools.	Highly Important	Medium	SMM
Develop scalable power monitoring and profiling tools for computational components, applications, and facilities.	Highly Important	Medium	SMM
Develop scheduling and resource allocation strategies to utilize component and application power profiles in scheduling, allocation, and power management decisions.	Highly Important	Hard	SMM
Develop tools to perform power profiling of HPC systems and their application workloads at a physical plant level.	Highly Important	Easy	SMM
Develop tools for automated processing of all available logs.	Highly Important	Hard	SMM

Strategy	Priority	Difficulty	Technical Area*
Develop an extensible set of APIs targeted at enabling external interaction with monitoring and analysis services.	Highly Important	Hard	SMM
Develop scalable, platform-independent data sampling tools and methodologies that provide time synchronous samples across all monitored objects to produce a system "snapshot" capability.	Highly Important	Easy	SMM
Develop scalable parallel storage and retrieval capabilities for sampled data where both in-band and out-of-band data are stored to targets accessible by analysis tools.	Highly Important	Easy	SMM
Develop scalable analysis tools that utilize both numeric and log file data to perform both run-time and post-run analyses and expose results via an appropriate API.	Highly Important	Medium	SMM
Develop and/or deploy scalable compression techniques for moving old data to long-term "cold" storage. This data needs to be retrievable for historic analysis purposes. Dedicate appropriate long-term storage resources to maintain this data at least over the life of the machine (~100 PB or more uncompressed).	Highly Important	Medium	SMM
Develop methods that enable a combination of exploratory (human) analysis and in situ techniques.	Highly Important	Medium	DAV
Develop tools to specify, modify, and deploy complex workflows in an efficient and intuitive manner.	Highly Important	Hard	DAV
Develop techniques to enable the use of partial ensemble analysis results to inform and direct job-management frameworks (for example, Dakota).	Highly Important	Medium	DAV
Develop standard methods for data management and provenance.	Highly Important	Hard	DAV
Invest in the creation of in situ components that can be leveraged by domain scientists for inclusion in their codes.	Important	Hard	DAV
Research in-transit techniques to leverage the increasingly deep memory hierarchies of future systems.	Important	Medium	DAV
Perform R&D for in situ data reduction.	Important	Medium	DAV
Perform R&D for in situ anomaly detection and feature extraction.	Important	Medium	DAV
Perform R&D for data management and exploration.	Important	Medium	DAV
Perform R&D for visual analytics of ensembles of runs.	Highly Important	Hard	DAV
Design with productization in mind.	Essential	Easy	DAV
Maintain tight integration with IC program element.	Essential	Easy	DAV

Strategy	Priority	Difficulty	Technical Area*
Collaborate with external developers and vendors.	Important	Medium	DAV
Migrate existing hardware resources into long-term infrastructure maintenance and renewal programs.	Important	Easy	DAV
Ensure evolutionary transition to new usage models.	Essential	Easy	DAV
Include consideration of visualization and analysis hardware as part of the platform procurement.	Essential	Easy	DAV
Include dedicated visualization resource near the compute platform.	Important	Medium	DAV
Include appropriate local rendering and capability visualization and analysis resources in site planning; coordinate as needed across the sites.	Highly Important	Medium	DAV
Upgrade performance of local hardware over time, consistent with platform capabilities and applications requirements.	Highly Important	Medium	DAV
Continue investment in interdisciplinary efforts and co-design.	Highly Important	Medium	DAV
Define and impose requirements on networking, file systems, and storage areas to support remote access and to support access to the file system and other local compute resources.	Highly Important	Medium	DAV
Continually stay abreast of commodity hardware and software developments and investigate ways in which software and algorithms designed for such hardware could produce benefits that could be applied to ASC exascale needs.	Essential	Hard	DAV
Leverage collaborations and research investments.	Highly Important	Medium	I/OFSS
Integrate storage system and I/O resources into HPC schedulers.	Highly Important	Hard	I/OFSS
Investigate, design, and deploy hierarchical storage solutions.	Highly Important	Medium	I/OFSS
Implement Trinity and Sierra burst buffer.	Essential	Medium	I/OFSS
Design and construct integrated analysis code suites.	Important	Medium	I/OFSS
Expand PFS into NVM.	Important	Medium	I/OFSS
Leverage key-value interfaces for burst buffer.	Important	Easy	I/OFSS
Ensure shared priority for at least a portion of burst buffer resources for other uses.	Important	Easy	I/OFSS
Include write count as part of cost.	Important	Easy	I/OFSS

Strategy	**Priority**	**Difficulty**	**Technical Area***
Allocate number of writes as part of compute award.	Important	Easy	I/OFSS
Deploy alternatives to traditional RAID.	Highly Important	Easy	I/OFSS
Leverage robust networking middleware.	Highly Important	Hard	I/OFSS
Research, develop, and deploy distributed namespaces.	Important	Hard	I/OFSS
Accelerate small, unaligned data transfers and metadata operations.	Important	Medium	I/OFSS
Investigate alternative metadata organizations.	Highly Important	Hard	I/OFSS
Eliminate POSIX bottlenecks.	Highly Important	Hard	I/OFSS
Leverage advancements in checkpoint algorithms.	Important	Medium	I/OFSS
Develop intermediate storage tiers—Campaign Storage.	Highly Important	Medium	I/OFSS
Expand HPSS disk caches.	Highly Important	Easy	I/OFSS
Add tape ordered recall and mount awareness in HPSS.	Important	Easy	I/OFSS
Develop multiple metadata servers in HPSS.	Highly Important	Medium	I/OFSS
Encrypt HPSS interfaces and data at rest.	Important	Medium	I/OFSS
Archive scrubbing and automatic checksums.	Important	Medium	I/OFSS
Perform faster development of high-speed type I IP encryptors.	Highly Important	Hard	NI
Increase reliability of encrypted WAN communications system.	Essential	Medium	NI
Scale network protocols to maintain efficiency for pre-exascale.	Essential	Medium	NI
Develop techniques for new bulk bandwidth technologies.	Important	Medium	NI
Develop technologies that maximize effective bandwidth of network interface cards (NICs).	Important	Hard	NI
Develop techniques for storage area network interconnect management technologies.	Important	Medium	NI
Partner with industry to engage in cutting-edge research and development in the area of high performance interconnects.	Highly Important	Medium	NI

Strategy	**Priority**	**Difficulty**	**Technical Area***
Develop performance modeling and analysis tools to judge the impact of changes to key architectural features.	Highly Important	Medium	NI
Place higher emphasis on HPC network requirements in future acquisitions.	Highly Important	Medium	NI
Develop management and performance analysis tools for compute interconnect.	Important	Medium	NI

* key to technical areas:

PET	Programming Environment and Tools
SMM	System Management and Monitoring
DAV	Data Analysis and Visualization
I/OFSS	Input/Output, Files Systems, and Storage
NI	Networks and Interconnects

Appendix D. Level 2 Milestone Text

<table>
<tr><th colspan="3">Milestone (ID#5216): Planning for Pre-Exascale Platform Environment</th></tr>
<tr><td>Level: 2</td><td>Fiscal Year: FY15</td><td>DOE Area/Campaign: ASC</td></tr>
<tr><td colspan="3">Completion Date: 9/30/15</td></tr>
<tr><td colspan="3">ASC nWBS Subprogram: CSSE</td></tr>
<tr><td colspan="3">Participating Sites: LLNL, LANL, SNL</td></tr>
<tr><td colspan="3">Participating Programs/Campaigns: ASC</td></tr>
<tr><td colspan="3">Description: This milestone addresses planning for hardware architectures and associated system environments anticipated in the 2016–2021 timeframe for advanced technology platforms. The study will be strongly aligned with the concurrent development of the usage models for the ATS-1 (Trinity) and ATS-2 (Sierra) platforms. Given the ATS-1/2 baseline(s), user and application environment requirements for future ATS will be considered. A review of the applicable R&D landscape will be performed, followed by a gap analysis, and finally a next-steps proposal. Known areas for attention include: strategies for application porting, readiness and scaling, based on expected programming models and tools for application development/execution; strategies for maximizing I/O efficiency as part of an overall computational analysis workflow; strategies for delivering system performance, reliability and associated resource management that enables real use of such systems at their intended capability; strategies for maximizing whatever energy-efficient utilization can be achieved for what are expected to be expensive systems to operate. The ability to deliver next-generation platform environments that provide an effective, efficient computing ecosystem in support of the ASC mission is the driver for this work.</td></tr>
<tr><td colspan="3">Completion Criteria: A report covering the strategy to provide the platform environment components needed for successful ASC use of advanced technology platforms.</td></tr>
<tr><td colspan="3">Customer: DSW customers and ASC IC and ATDM subprograms</td></tr>
<tr><td colspan="3">Milestone Certification Method:
Professional documentation, such as a report or a set of viewgraphs with a written summary, is prepared as a record of milestone completion.
A program review is conducted and its results are documented.</td></tr>
<tr><td colspan="3">Supporting Resources: Tri-lab CSSE staff with input from IC and FOUS staff.</td></tr>
</table>