Appendix 1

ExaRD Detailed Technical Descriptions

Contents

ExaRD Detailed Technical Descriptions

The challenges of exascale computing demand holistic, comprehensive solutions in computer hardware, software environments, and advanced algorithms. ExaRD aims to deliver these comprehensive solutions, embodied in the ECI Software Ecosystem.

The ECI Software Ecosystem will provide a comprehensive software framework and environment for the development, test, evaluation and maturation of exascale software components. The ecosystem will consist of components originating in the vendor, national laboratory, academic research and open source communities. To the maximum extent possible the ecosystem will consist of components common to all ECI vendor platforms. For components that must be vendor specific, the ecosystem will serve as a software test bed to develop modular approaches with well-defined, common APIs. The ECI Software Ecosystem will consist of Operating Systems and runtime systems; programming models, compilers, and libraries (including math libraries); debug and performance tools; system management and resource allocation tools; data analytics, visualization, and other high-level tools; and other specialized software as required. For each vendor exascale research platform, the software ecosystem will initially consist of a basic software suite of essential vendor-supplied components augmented by any third party software, open-source software, and laboratory-supported software required by the application community. Software from the research and vendor development communities for managing and utilizing new hardware and software capabilities and concepts will be merged into the ECI Software Ecosystem for testing in a prototype production environment.

To address the exascale challenge, the ExaRD component contains following technical areas:

1. **Application Foundations (AF):** Research, development, and integration of software technologies for the foundation of future exascale applications. Sub-areas include Co-Design (CD), Applied Mathematics (AM), and Data Analysis and Visualization (DV).
2. **User Experience (UE):** Research, development, and integration of software technologies that enable the utilization of the computer system. Sub-areas include Resilience (RE), Productivity (PR), Collaborative Environment (CE), Cyber Security (CS) and Application Integrity (AI).
3. **Software Stack (SS):** Research and development in software technologies such as operating systems and runtimes, as well as programming environments and tools.
4. **Performance Execution (PE):** Independent, quantitative analyses, models, and predictions of application performance on future systems.
5. **Data Management (DM):** Infrastructure for data storage and movement.
6. **Hardware Architectures (HA):** Hardware architecture information distillation and dissemination, design of alternative system.
7. **System Engineering and Integration (SI):** integration of computing technologies from ExaRD and industry to ensure that the overall exascale system design meets DOE mission needs and facility requirements.

The following sections provide detailed information for the ExaRD technical areas.

1 Application Foundations

The goal of the Application Foundations area is to solve DOE mission-critical problems at unprecedented scale, with an unprecedented level of fidelity. The Application Foundations area will ensure that the requirements of the applications are met and that the applications run efficiently on the eventual exascale machine.

The Application Foundations area has three components: Co-Design, Applied Math, and Data Analytics and Visualization.

1.1 Co-Design

The Co-Design technical area will investigate the algorithmic, hardware, and software requirements and the associated trade-offs, in the context of relevant applications. As the primary provider of integrated use cases, Co-Design drives fundamental, focused research in hardware, software, algorithms, and application modeling. Co-Design in turn will engage the science and engineering application domain experts, hardware architects, mathematicians, computer scientists, data specialists, and others, to integrate and adapt suitable combinations of available technologies to solve the application problem at hand. Thus the Co-Design Centers are expected to coordinate multiple interactions and will need to devote significant effort to communicating application development requirements and lessons learned to the developing exascale computer systems. In the process of integration, Co-Design projects will learn about future technologies in hardware, software, and algorithms, and the requirements they place on the new applications being developed. The Co-Design Centers are expected to be flexible and agile with frequent reviews to ensure that projects remain on the path to success.

The proposed projects for this technical area are described in the following sections:

CD1: Application Co-Design

This project will fund the establishment of several Co-Design Centers. Co-Design Centers are envisioned to be large, integrated projects. Each is *anchored* in a chosen application area (such as combustion, materials science, or nuclear reactors simulation) with the goal of developing the *computational infrastructure* that will enable domain scientists and engineers to deploy exascale computing technologies efficiently. The targeted applications are chosen to be relevant to critical Department missions. The project team is selected to have extensive experience in application code development and scaling, in deploying a wide array of software and hardware tools to gain performance, and in working effectively in large interdisciplinary collaborations.

The principal objectives of the Co-Design Centers are:

- Define and analyze computational requirements associated with workflow of key applications, and predict how those requirements map onto proposed exascale hardware;
- Determine desired characteristics of the software stack and programming environment needed to implement and execute computational models;

- Communicate applications and algorithms requirements to the exascale platform developers, and to applied mathematicians and computer scientists who work on algorithms; and
- Work, in concert with stakeholders, to analyze the trade-offs of choices in these areas.

Co-Design Centers will provide *use cases* for research and development supported by the ECI. The Co-Design Centers will release a suite of "proxy applications" that embodies key characteristics of the application workload, such as communication and data access patterns, memory requirement, function calls, etc. Co-Design Centers may also work with interested parties to tailor proxy apps that will focus on particular properties of interest. The Co-Design Centers will document the computational, data movement and data storage requirements for exascale computing platform developers. The Centers are not expected to produce production-level codes; it is nevertheless expected that each Co-Design Center would release a suite of research codes that represent the body of knowledge gained during the project, and that guide future production code-development efforts.

CD2: Cross-cutting Co-Design

Projects will explore specific topics that cut across all the Application Co-Design Centers. The cross-cutting research projects are intended to be in-depth, interdisciplinary research conducted in the spirit of "co-design." Whereas Application Co-Design Centers develop breadth, these cross-cutting research projects provide the opportunity to develop in depth what Application Co-Design Centers cannot afford to. For example, by co-designing a programming environment or a domain-specific language (DSL) with algorithms, instead of researching DSL for fixed algorithms, the computer scientists working on the DSL gains insight into how the algorithms should do, while the mathematicians working on the algorithms can see the limitations of the programming environment and strategize accordingly. These interactions allow for more in-depth studies of programming environments, algorithms, tools, etc., and are appropriate for continuing promising directions from an Application Co-Design project.

CD3: Tools and Knowledge Base

This project provides the means to document the results of Co-Design programs that facilitate future Co-Design activities, even after an Application Co-Design project has ended.

This project will support multiple, low-level, on-going, focused efforts whose specific aim is to capture research artifacts from the Co-Design programs by developing production-level tools and maintaining a knowledge base. The emphasis here is on results arising from the Co-Design activities that may be enable efficient use of exascale machines. For example, this includes tools that allow for automatic workflow analysis whose output may be communication patterns, memory access patterns, domain decomposition analyses, numerical accuracy requirements, and other important characteristics of an application that make up the typical workflow, as well as what the characteristics of the workflow may be.

The outputs of this project will provide crucial starting points for effective two-way communication between Co-Design projects and vendors, DoE labs, universities, and other parts of the Exascale Ecosystem. This project aims to capture lessons-learned and will reduce future duplication of work by having a good starting point with effective tools.

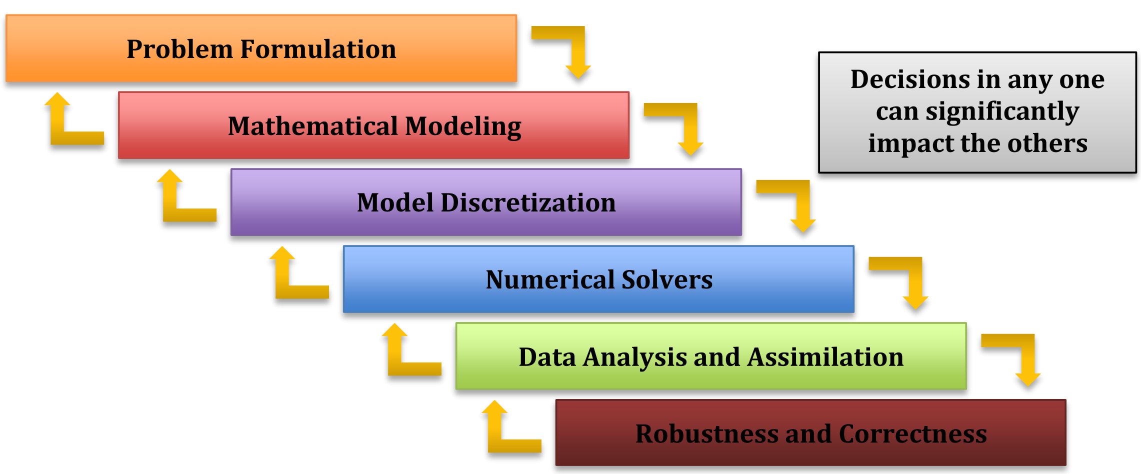
CD4: Technology Transfer

The Exascale Computing Initiative is not borne out of the desire for a "blue sky" or "moon shot" project but the economic necessity of being able to carry out National missions at a sustainable cost. It is anticipated that the 20MW power envelop may require disruptive technologies, and that these technologies may have profound impact on the designs and operations of devices ranging from cell phones, laptops, or whatever other future personal computing devices, to small and medium computing devices used in businesses of all sizes, to data-acquisition, processing, and storage systems, to large high-performance computing systems. With these profound changes come the corresponding changes in software and applications. The "spin off" is an important measurement of the success of the Exascale Computing Initiative, and the Co-Design Centers play a key role in this area.

1.2 Applied Mathematics

The overarching goal of the Application Foundations Applied Mathematics project is to coordinate and foster advances in mathematical models, analysis, algorithms, and software to enable extreme-scale science. The promise of exascale computing is that it will not only provide more computational resources enabling higher-resolution simulations and more demanding studies, but also enable the community to pose new scientific questions. The disruptive exascale architectural characteristics, which will introduce new concerns and change the criteria for which algorithms are designed, will inform the redesign of existing and invention of new algorithms and numerical software libraries. The Applied Mathematics project will rethink, reformulate, and develop new mathematical techniques to deliver more complete and sophisticated models and new predictive simulation and analysis capabilities.

Analogous to the concept of the *software stack*, there is effectively a *mathematics stack* for simulation and analysis:



Decisions made in any one of these mathematics stack activities can have significant effects on the others. While subsets of the various issues are sometimes approached in an integrated fashion, better end-to-end integration of the stack is necessary.

Beyond simulation and analysis, the Applied Mathematics project will provide:

- Optimization and optimal control approaches for *system management*, e.g. the *Software Stack* and *Systems Engineering Integration*
- Data reduction, data analysis and transformation algorithms for *Data Analytics and Visualization* and *Data Management*
- Stochastic performance models for *Performance Execution*
- Libraries and APIs that contribute to *Productivity*

The exascale Applied Mathematics project involves three complementary elements that will advance exascale mathematics on three fronts: from a transformative perspective that extends our existing numerical software libraries as far as possible into the exascale; from an integrated, end-to-end perspective across the mathematics stack; and from a more focused research perspective that invents new models and algorithms better-suited to take advantage of exascale architectures.

The proposed programs for this technical area are described in the following sections:

AM1: Advanced Numerical Software Library (ANSL)

Applied research, development, and support is needed to extend the large collection of existing DOE mathematical software libraries and frameworks to make better use of exascale architectural features. Software libraries are powerful means of sharing verified, optimized algorithms, and, accordingly, the DOE has invested in the development of numerous numerical libraries. Many applications rely on these numerical libraries, and transforming these libraries for exascale involves research and development, not just re-implementation. Furthermore, as exascale opens up opportunities for new analyses such as optimization and uncertainty quantification at scale, algorithms that can be generalized should be made available through new libraries and frameworks.

As such, investments in existing and new mathematical software must continue. Within ANSL, the research focus will be on transforming existing algorithms to perform as well as possible on next-generation architectures, in terms of both on-node and inter-node parallel performance. Entirely new algorithms will mostly be developed under the Mathematics Basic Research projects (described below), but their development into hardened, verified, and supported software libraries will occur within ANSL. Reasoning about the performance and trade-offs on such systems will be more difficult, thus optimal configurations will be more easily discovered than predicted. Numerical software libraries must also be made resilient to faults. At the very least this will require interfacing with APIs provided by the resilience framework operating across the software stack. This project includes the resources to help support such ancillary activities and capabilities.

AM2: Mathematics Integrated Research Centers (MIRC)

The mathematical areas represented in the *mathematics stack* should be considered holistically since there are interdependencies throughout the stack. Choices made in formulations, models, and discretizations, for instance, constrain the possible solution methods and parallel implementations, while the availability of scalable solver algorithms places limits on problem formulation and model choices. For example, discrete models inherit properties of the models they approximate (e.g., propagation speeds) and discretizations dictate the structure of the resulting global (non)linear algebra problem. Thus, when designing exascale applications, there should be advantages to considering the mathematics as a whole, from the goal of the computation and the problem formulation through to how the discrete approximation is solved. Furthermore, by considering the whole problem, there may be opportunities to alter traditional solution strategies and re-order operations. For instance, by designing uncertainty quantification into a solution process from the outset – instead of as an outer loop around a forward solver – one could take advantage of increased concurrency within the inner loops by solving for groups of nearby solutions simultaneously. Another advantage may be in resilience; considered as a whole, the layers of and properties of the mathematics stack could work together to implement a minimal but sufficient detection and recovery scheme.

Such a *mathematics co-design* approach is necessary. This is separate but related to the Co-Design activities, which are more broadly focused from the application through to the hardware. The MIRCs will be integrated mathematics (across the mathematical sub-disciplines) research projects that will make extreme-scale hardware and software challenges a central focus in their holistic approach. The goal is to develop mathematical approaches that balance the needs and constraints throughout the mathematics stack with a focus on specific, DOE-relevant grand-challenge problems, including problems of specific interest to the ASC and Office of Science programs as identified by the Exascale Computing Initiative. Within each center, problem formulation will be informed by these DOE science problems and will holistically consider multiscale, multiphysics models; optimization for design; inverse problems; and uncertainty quantification, as appropriate.

AM3: Mathematics Basic Research Projects (MBR)

The move to exascale will be a more disruptive transition than the previous moves to terascale and petascale because of the significant changes in computer architecture. Current algorithms will need to be evaluated and either modified or abandoned; alternative algorithms will need to be invented or resurrected. The promise of exascale resources will also provide opportunities for new or alternative models and problem formulations, but how and when to use such models are open questions. There is much investigation to be done around specific classes of models and algorithms. Because of the complexity of the problem and the numerous trade-offs, it is too early to designate any one technology or approach as superior, and so a diverse research portfolio is the most robust investment strategy at this point. The MBR project, designed to range in scope from one to several areas of the math stack, is intended to engage a broad range of applied mathematicians.

In contrast to the MIRC approach, the MBR projects can range in scope from one to several areas of the math stack. Research areas under consideration include uncertainty quantification; optimization, including mixed-integer constraint optimization and optimization under uncertainty; new formulations and exascale-suitable discretizations; parallel-in-time discretizations; high-order discretizations; mesh, model, and discretization adaptivity; advanced multi-physics and multi-scale solution techniques including partitioned time integration; novel solver algorithms including communication-avoiding, synchronization reducing, randomized, multi-precision; scalable computational geometry and mesh generation; data analysis algorithms; new approaches to complex code verification and reproducibility within the context of less reliable computer systems; and advanced optimization, statistical techniques, and discrete mathematics to improve system resource management, load balancing. Within any of these areas, algorithmic-based fault tolerance techniques can be considered in coordination with the Resilience team.

1.3 Data Analytics and Visualization

Data analytics and visualization are processes that enable detection, analysis, and visual representation of patterns in data, aimed at supporting knowledge discovery and decision making. Data analytics refers to the process of transforming data into an information-rich form via mathematical or computational algorithms to promote better understanding. Visualization refers to the process of transforming scientific simulation and experimental data into images to facilitate visual understanding.

The DV technical area is focused on the research, design, implementation, deployment, and application of data analytics and visualization software suitable for use on exascale platforms.

The current data analysis and visualization workflow is primarily executed *post-hoc*, relying on availability of persistent storage to which the data may be exported. On exascale systems, however, the power cost of data movement and the worsening input/output (I/O) bottleneck will make it necessary for simulation data to be analyzed *in situ*, or on the supercomputer while the simulation is running. Software for data analysis and visualization must be refactored or even re-architected with consideration to characteristics of exascale architectures.

The proposed programs for this technical area are described in the following sections:

DV1: Foundational Research

Research and advanced development is required in the following key topics:

- In Situ Processing Methods and Infrastructure
- Data Reduction and Post Hoc Data Analysis and Visualization
- Data Analytic and Visualization Methods for New Science
- Experimental and Observational Data
- Data Analytic and Visualization for Exascale Technology R&D
- Data-Centric Supercomputing

DV2: Data Analytic and Visualization Algorithms

The traditional approach for implementing key data analytic and visualization software – codes based upon explicit parallelism using thread- and MPI-based constructs – will not scale into the exascale regime, nor provide the flexibility and adaptability needed for exascale-class applications, such as being able to leverage advances in optimized runtime systems, resilience, or power efficiency.

This project aims to revolutionize data analytics and visualization software architectures and implementations to achieve extreme levels of concurrency, to be readily portable across platforms, to be economical in use of power, have better reliability, to increase the sustainability and lifespan of data analytics and visualization software, to enable use of novel architectural characteristics, and to take advantage of big data science developments from diverse research programs at the labs, academia, and industry.

It is envisioned that the development of an environment where high-level, science-facing data analytics and visualization applications and tools with specific capabilities needed by science are constructed using data analytics and visualization algorithms, which are in turn built upon data-parallel libraries and programming models optimized for key data-centric motifs. This design accommodates the interactions with many other ExaRD technical areas, as well as creates opportunity for broadly leveraging developments from industry and research, and promotes the sustainability of data analytics and visualization software for the future.

2 User Experiences

The goal for the User Experiences technical area is to address topics that support scientists' ability to use, interact with, and understand the exascale system and the quality of their experiences with and scientific outcomes from the system. Focus areas include human computer interaction, data analytics and visualization, collaboration environments, productivity, and integrity (i.e., resilience, cyber security, and program validation).

The User Experiences has three components: Productivity (PR), Collaborative Environments (CE), Resilience (RE) and Application Integrity (AI).

2.1 Productivity

The goal for the Productivity technical area is to conduct research and development to identify, ameliorate, and remove potential productivity bottlenecks in order reduce the overall time to solutions for exascale applications. Here "time to solution" includes two components: application software development time (development time productivity), and application software execution time (execution time productivity). We define *application software development time* as the time it takes to specify an physics/model requirements, transform the requirements into software, compile and test the resulting software, and make it ready for execution on a target hardware system, and *application software execution time* as the total time that it takes an application code to execute its workflow to completion on a target machine and produce the desired scientific results. In the context of time to solution, productivity will play a major role in the lifecycle of an exascale system, from research and development to operation, enabling the system to deliver increased value to users at a rate commensurate with the performance of the underlying hardware technologies.

The proposed programs for this technical area are described in the following sections:

PR1: Productivity Foundations

The optimal utility of an exascale system is determined by a complex set of factors related to performance; programmability, efficiency, and system availability that ultimately determine the rate of science deliverables are produced. Yet there is no community-wide measure that captures the essential crosscutting objective function reflecting the combined effects to rationally guide optimization. This project will derive the foundations of "productivity" as a useful quantitative abstraction to guide all aspects of computational science and to develop the necessary tools for practical measurement and application. Project activities include

- Develop methodologies for measuring productivity, including metrics and benchmarks for productivity
- Derive a formal representation of productivity in terms of principal parameters to permit sensitivity analysis
- Determine the effect of each of the system component layers for overall system productivity to derive their interrelationships and guide interoperability

PR2: System Productivity Tools

The goal of this project is to develop tools, methodologies, and processes to improve both development time productivity and execution time productivity. This project will consist of two major tasks:

- Tools for development time productivity: Here we will enable efficient programming of reusable, maintainable, and portable codes across different architectures without sacrificing simulation fidelity and performance, for the entire scientific workflow. Activities will center on the adoption and enhancement of software engineering practices and processes for transforming legacy codes, interfaces to math libraries, and integration and adaptation of programming tools and environments
- Tools for execution time productivity: Here we will enable the efficient utilization of systems' resources to minimize execution time of application workflow. Activities will focus on the development of holistic tools that allow the modeling, monitoring, and optimization of energy expenditure, resource utilization, scalability, and time to completion of application runs

These two tasks will involve extensive crosscutting and collaborative activities across the entire exascale ecosystem.

2.2 Collaborative Environment

Extreme scale science is ushering in an era of collaborative knowledge discovery where teams of scientist will work together in *Collaborative Interactive Exascale Computing* (CIEC) environments to explore complex issues from multiple points of view. Some collaboration will expand existing computationally expensive simulations by exploiting multi-user support to the analysis and visualization systems. In other cases the science team will interactively explore their experimental and observation data and exascale simulations in order to further their scientific understanding. The common thread that makes any collaboration successful is the ability of the team members to interact with each other, their data, and their workflows, enabling timely decisions to be made to advance towards a common goal.

The science communities that motivate this project are working to exploit opportunities inherent in combining exceptionally powerful scientific instruments, including light sources, tokomaks, microscopes, telescopes, accelerators, and genome sequencers with exascale supercomputers. Data growth is driven by both the rapid advancements in instrument capabilities such as number of detector pixels, image sizes, refresh rates, and the increased computational power used to generate simulations. CIEC will develop the essential technologies that will enable the timely analysis and use of data generated by both the experimental and observation data and computational science communities. The common challenge faced by all these communities is processing their data within timeframes that permit useful decision to be made. CIEC will be essential in fully developing these team-based science collaborations.

The CE technical area will enable the Collaborative Interactive Exascale Computing environment required to support the multi-user interactive exploration of data.

The proposed programs for this technical area are described in the following sections:

CE1: Rapid Data Manipulation and Processing

Experimental scientists often need rapid access to a subset of the experimental data in order to find out if enough data has been collected to allow scientifically valid analysis. Computational scientists will also benefit from rapid access to a subset of the simulation. They would also benefit from knowing when the simulation has stopped making progress. These decisions can only be made if the data can be quickly processed and results presented back to the scientist in sub-minute time frames.

This project will develop the tools and frameworks needed to extend existing computer-in-the-loop decision processes to enable interactive human observation of this data. It will develop new mechanisms that will enable the system to compare the new data with archived data to determine if an experiment is worth continuing. Finally, it will develop the services needed to automatically take corrective action to detect when experimental devices are beginning to exceed their design limits. Collaborative groups will do this work from academia, industry, and the labs.

CE2: Multi-user Complex Workflows

Teams of scientists work collaboratively to explore complex issues or to bring multiple viewpoints to bear to solve a complex problem. Exploring experimental data, simulation data, or a combination of both may do this analysis. Given the complexity of modern science, it is no longer feasible to expect that every team member will be aware of all the existing research results and hypothesis in their field. Advanced multi-user decision taking workflows may be developed to rapidly assemble the required data sources, analysis tools and visualization systems needed to assist the team in exploring this problem. This gathered knowledge, coupled with the automated assessment of the emerging results, will enable research teams to validate existing theories, interpret new data, and develop new avenues for future research.

This project element will develop the theories, tools, and services needed to automate the creation of these decision-making workflows. It will enable multiple scientists to simultaneously and rapidly explore different views of a complex problem. The workflow environment will be able to automatically expand and contract to meet the rapidly changing needs of the experiment and analysis tasks. Multiple scientists will be able to interact with their peers and with the applications running on an exascale computer.

CE3: Interaction Services

Collaborative environments need to be able to support the “alignment of mental models” among experts in diverse domains. Given the scope and complexity of scientific analysis tasks or the detail found in the visualized images, human senses need to be augmented with information extracted from knowledge repositories if researchers are to reach valid or insightful conclusions in any reasonable timeframe.

This project will create the methods and tools needed enhance the scientist’s ability to interact with their data and the data analytics and visualization systems manipulating this data. It will research mechanisms that reduce the time and expertise needed to use advanced workflows effectively. It will research mechanisms that scale existing single-user tools and services to meet the needs of multiple simultaneous user communities. As a result, new methods for supporting collaborative data exploration, workflows, decision taking, and hypothesis testing among teams of scientists, both co-located and distributed, are needed. Collaborative teams of academic and lab researchers will perform this work.

CE4: Knowledge Management and Validation Mechanisms

Next-generation discovery systems will require mechanisms that effectively manage the large quantities of diverse information, domain concepts, and data from simulations, experiments, and models. It may also require the integration or fusion of such data and knowledge into consistent, accurate, and useful representations of real-world objects. Finally, the results, evidence of their validity, as well as decisions taken to generate these results, should be captured in a (machine readable) knowledge base. This project will create the tools, mechanisms, and services needed to capture, evaluate, retain, and recall all the information needed to reproduce an experiment’s results. Integrated mechanisms that enable the capture, evaluation, and display of curation and provenance data at high velocities in computationally rich environments need to be created. Scalable mechanisms that automatically transform and integrate data from multiple disparate sources at runtime are also required to meet these new science demands. Collaborative teams of university and lab investigators will carry out this work.

2.3 Resilience

The overarching goal of the Resilience technical area is to keep the application workload running to a correct solution in a timely and efficient manner on future systems even in the presence of increasing failures, challenges in I/O scalability for checkpoint/restart and silent (undetected) errors.

The Resilience technical area will have ongoing interactions with application teams, leadership computing facilities, system designers, algorithm research teams, and all the other ExaRD efforts to develop a comprehensive understanding of the requirements, capabilities, and gaps to having resilient applications and systems. The Resilience project will use these interactions to help coordinate R&D efforts, and to facilitate integration and compatibility of hardware and software resilience solutions. The research priorities for the Resilience project in the exascale timeframe are fault characterization, fault detection, fault-tolerant algorithms, fault-tolerant programming models and tools. Metrics for success include the demonstration of practical application resilience strategies on pre-exascale systems and eventually the exascale systems.

Advances in resilience require and benefit from a particularly diverse set of activities. Improved reliability in one layer of the system stack can have strong positive impacts on other layers and applications, reducing the required efforts in other areas, or permitting simplified designs and implementations. Increased degradation in an area can have similar negative effects across the stack.

The proposed programs for this technical area are described in the following sections:

RE1: Capabilities, Requirements and Gaps

The goal for this project is to regularly reevaluate the exascale resilience requirements, capabilities and gaps by having annual (or more frequently if needed) meetings with each of the following groups:

- Application Teams
- Computing facilities
- System designers
- Algorithms developers

The Resilience projects will gather requirements regarding resilience needs and inform their efforts with regard to application and facility resilience needs.

In addition to regular communication between the Resilience projects and Application Foundations Managers, an annual report will be produced detailing the latest resilience requirements, capabilities, and gaps. One risk in this project is that the system designs and technology changes may cause new types of errors to emerge, creating new gaps and forcing applications to rethink their algorithms and fault mitigation strategies.

RE2: Coordinate Resilience R&D

Resilience research and development is very diverse across industry, academia, and the DOE labs. It is done at all layers of the software stack from the low-level hardware all the way up to the application workflow. These solutions are often developed independently of each other and perhaps for different purposes. To maximize the effectiveness of the different resilience efforts it is important to have a group coordinate the integrated effort.

RE3: Priority Resilience Research Topics

The Resilience effort is responsible for overseeing projects in topics not covered elsewhere, which includes fault characterization, fault detection, fault-tolerant programming models, fault-tolerant algorithms, and fault-tolerant/fault-aware tools. The following topics will be included in this project:

- Fault Characterization
- Fault Detection
- Fault-Tolerant Programming Models
- Fault-Tolerant Algorithms
- Fault-Tolerant and Fault-Aware Tools

This work will yield a variety of software tools that facilitate DOE's resilience research and development agenda.

2.4 Application Integrity

This research topic will seek to investigate and develop solutions for the protection of computation integrity of codes, services, and applications executed on the potential ECI hierarchical and heterogeneous high performance (HPC) computing platform. The challenge to address in this research topic is the assurance that during computation, the execution of concurrent codes, services, or applications cannot interfere or affect the computed result in an unquantifiable manner. Today, formal (software or hardware) verification techniques are applied to most system components before they are integrated and deployed in production systems. Commonly, the verification of these systems is done by providing a formal proof on an abstract mathematical model of the system, the correspondence between the mathematical model and the nature of the system being otherwise known by construction. Because of the potential of subtle interactions between components, it is increasingly difficult to exercise a realistic set of possibilities by simulation. Therefore, formal verification is commonly applied to limited sub-systems leaving room for potential vulnerabilities once these are integrated within the production environment. In some respects, research and development efforts in this particular topic may investigate the mechanisms to deliver effective solutions for performing *Real-Time Result Verification* within the constraints provided by the ECI system architecture such as processor memory, input/output bandwidth, storage, hierarchies, and communication network to name a few.

2.5 Cybersecurity (CS)

The main research challenge with cybersecurity is the ability to effectively detect and characterize anomalies that may register as faults or may not even disrupt the normal system operation. Another challenge is how to accurately determine if a perceived anomaly is naturally occurring within the system or maliciously caused. Success in cybersecurity today lies in the detection of signatures that are accurate but only detect what is known, and the application of anomaly detection algorithms as a post-processing analysis step with high levels of false-positive alerts. Both approaches assume full access to the data as most data are archived and later retrieved for analysis, if time and human resources permit. ECI presents new challenges to cybersecurity that combine (1) large amounts of data and information some of which cannot be saved or stored for later analysis, (2) limited computing and memory capacity for real-time monitoring during operation, (3) higher rates of faults and resilience issues, and (4) a more complex heterogeneous system architecture than currently exists. The goal for cybersecurity research and development within ECI includes the understanding of the attack surface area for the computing platform, network, data repository systems, and the elements that regularly access the system over various networks. In addition, research and development performed in the areas of resilience and computational integrity will lead to effective monitoring and reporting technology solutions that should seek to minimize any security overhead allowing the system to fulfill its scientific mission unimpeded.

3 Software Stack

The focus of the Software Stack technical area is the research, coordination and integration of the software needed to develop and execute applications on future extreme-scale platforms. A key goal of the software stack effort is to enable *performance portability* – the ability for exascale applications to achieve high performance without resorting to platform-specific methods.

The unparalleled concurrency and heterogeneity of computing components anticipated in exascale platforms, particularly at the node level, offers application developers ever more powerful capabilities to simulate and analyze complex phenomena. The never before imagined benefits, however, come with challenges also never before experienced. New programming models and environments will be needed for application developers to design and develop new codes with new algorithms and to refactor legacy codes that benefit from extreme-scale capabilities. New system software will be needed to manage power, resilience, heterogeneity, massive concurrency, and system variability, in order to achieve both high performance and low power on multiple platforms with possibly very different architectures. New tools to analyze and optimize performance, resilience, and power will also be needed to enable full utilization of architectural features to jointly maximize along these three dimensions.

The Software Stack team will coordinate research and development activities in innovative software technology. The major R&D activities are in four categories: (1) Systems Software; (2) Applications Support; (3) Development, Analysis, and Optimizations, (4) Exascale Software Ecosystem Integration.

The proposed programs for this technical area are described in the following sections:

SS1: System Software

System software is the lowest level of the software stack, and includes software designed to provide run-time layers and interfaces between applications, libraries, and the underlying specific hardware, as well as software used by system administrators and operators. Exascale system software must be designed to detect and respond to frequent failure of hardware and software components, as well as provide the interface between application support software and hardware. The projects will begin as extensions to many exploratory projects in the existing ASCR and NNSA research portfolios. Efforts include:

Operating System R&D: Future operating systems need to be able to globally and actively optimize for power, correctness, and resilience while providing the highest levels of efficiency possible. Optimal mapping of application executions onto system resources must be achieved in coordination with other parts of the tool chain, such as standard runtime libraries and resource managers.

Resource Management: In order to respond dynamically to system-wide power usage and faults, new resource managers must be developed with dynamic methods to manage deep memory hierarchies and a complex interconnection network to enable programmers to optimize data movement in their codes.

Infrastructure for Data Management: there needs to be system-level support for the use of future tools for data store and analysis, such as on-line filtering, in-situ processing, and offloading to special nodes.

SS2: **Application Support**

Application Support encompasses the research and development of the software that is directly used by application developers. The focus of the Application Support effort is to create usable programming environments that simplify application software development and refactoring for exascale platforms. These new programming environments need to support the current MPI based programming model, as well as future programming models. One example of new programming environments is a framework that accepts high level specifications from domain scientists using, for example, a domain specific language (DSL), enabling a key element for code portability and optimization: the separation of the specification of the algorithms from its schedule and its hardware-specific implementation and optimizations. A fully functional DSL tool chain will help code teams to more rapidly design, prototype, and deploy application codes with the guarantee that the mathematical properties of the original algorithms are fully preserved throughout levels of compilation, transformations, mappings, refinements and runtime optimizations, and thus ensure interoperability and portability across a variety of platforms.

As we search for significant productivity and performance portability enhancements, this project will also include programming language research and compiler development, with a focus on the *coordinated interoperability* of languages to allow incremental introduction of these technologies through libraries and reusable software packages. Existing compilers for languages widely used in scientific codes (e.g., C, C++, Fortran, and OpenMP) must continue to evolve to match capabilities of new hardware platforms and constraints imposed by power and resilience goals. The new programming environment may, for example, include tools to enable migration of current MPI+OpenMP codes to new extreme-scale software environments.

The Application Support R&D will focus on the coordination and co-design of front-end language features, refinements (e.g., source-to-source transformations, autotuning, and synthesis), and back-end optimizations such as polyhedral frameworks, with a goal of providing a consistent experience across multiple vendor-supported and open-source compilers.

SS3: Development, Analysis, and Optimization Tools

The ecosystem of tools, which are critical to the development and analysis of applications, such as debugging, and correctness tools, may need to be co-design with the system software and the application support efforts. One focus of this effort will be new scalable debugging techniques, extending lightweight techniques, exploring automated analysis through application behavior modeling and machine learning techniques. Another focus will be on tools that eliminate non-determinism in program executions and make concurrency bugs and floating point behavior reproducible. Tools that can measure timers, performance counters, and resource usage at all levels of the software stack will continue to be developed and designed with integrated data collection services. Leveraging innovations in data science for managing large sets of unstructured data, novel techniques to store, manage, visualize, and automatically analyze the data generated by performance measurement tools will also be researched. Testing and verifying correctness of the software ecosystem as well as of the application domain codes will become extremely difficult as scale increases. This effort will include developing and applying formal verification techniques to systems software and applications software, as well as the development of verification software tools.

SS4: Software Ecosystem Integration

The prototype software developed in the Software Stack technical area will be deployed for evaluation and integrated with software generated by ExaRD technical areas, the community, and vendors, requiring that a full system testing and maintenance strategy be developed.

4 Performance Execution

The Performance Execution effort will spearhead the research and development of the models and tools that provide independent, quantitative, and predictive evaluations of the performance of applications on future architectures. The Performance Execution technical area will also define requirements, baseline performance, identify performance metrics, develop a benchmark strategy, and make available results of benchmarking through databases to be accessed by all.

The results from the models developed by this effort will be used to monitor the progress towards achieving the ECI goals and to allow trade-offs to be made during this initiative.

For example, the results of the application modeling effort will allow to Co-Design teams to provide valuable advice to the computer vendors concerning the significance of design features.

The proposed programs for this technical area are described in the following sections:

PE1: Application Modeling and Simulation

In moving towards exascale systems, issues of data locality, concurrency increases, power constraints, overcoming faults, and systems and applications adaptivity will need to be tackled. Modeling and simulation can be applied to explore parameters in systems and applications designs in advance of implementation and assist in dynamic optimizations during system and application execution. Research topics include:

- Integrated Models & Tools for Performance and Power
- Component and system models for application performance
- Benchmarking Strategy
- Metrics for evaluating performance
- Dynamic Modeling

PE2: Execution Model Characterization

An execution model is an abstract layer that connects applications and algorithms with the underlying hardware and systems software through its semantics. The execution model can dictate the selection of algorithms, applications, program optimizations, and hardware designs. Execution models define the dynamic operations provided by the system hardware/software stack, which implement the runtime behavior expected by the application.

This effort will characterize, both quantitatively and qualitatively, potential new execution models in terms of performance, energy efficiency, and resilience, with the short-term goal of determining whether a new execution model is needed for exascale computing. This determination must be made in time to influence hardware architectural designs.

PE3: Productivity Modeling and Simulation

"Productivity" is often an ill-defined term, a mixture of the wishes, expectation, and the reality of computational science. Seemingly simple metrics such as "time to solution" may be an ill-understood function of ease of programming, debugging, time for compilation, resource allocation, etc., and ultimately actual time of execution on a machine..

This effort will support coordinated research and development of productivity modeling and simulation, and ensure that the tools and models be tested with realistic application workload. This is a relatively new research topic; the availability of petascale and exascale prototypes should provide important opportunities to explore ideas in modeling and simulation of productivity that may lead to greater understanding and clearer definition of productivity.

PE4: Lifecycle Modeling and Simulation

Modeling tools are key components of system and application design and behavioral analysis; modeling capabilities may play an integral role during all phases of both system and software lifecycle. Early in the lifecycle, modeling and simulation allows the exploration of feasible system designs and their impacts on application workloads. As a system is being built, models can be used to inform the performance at scale before the full system is available. Application-specific modeling techniques may verify if a system is correctly installed and configured, and point to opportunity for optimization, and hardware or application improvement. Deviations from expected behavior may be an indication of faulty operations. More important, having such a lifecycle tool may allow the prediction of future system behavior, thus guiding decisions within phases of the lifecycle.

This effort will consist of coordinated research and development in modeling tools to ensure state-of-the-art tools and practices can be deployed in systems design, acquisition, installation, operation, maintenance, and decommissioning, with the ultimate goal of producing usable tools to guide decisions made within System Engineering and Integration.

5 Data Management

The Data Management technical area will focus on developing the infrastructure necessary to manage, understand, and share data in support of DOE computational and data-centric science activities at exascale. The results of this effort will define capabilities that define how data are moved on/off the system, how certain types of data manipulation are performed within the system, and provide the infrastructure for checkpoint/restart, serving as the primary method of resilience to node failures.

Expected changes in the hardware architecture of exascale supercomputers will render current approaches to data management infeasible, resulting in disruptive changes to the scientific workflow and rendering traditional checkpoint/restart methods infeasible. A major concern is that exascale system concurrency is expected to grow by at least three orders of magnitude, yet system memory and input/output (I/O) bandwidth/persistent capacity are only expected to grow by one and two orders of magnitude, respectively. The reduced memory footprint per floating point operation further complicates these problems, as does the move to a hierarchical memory structure. The scientific workflow currently depends on exporting simulation data off the supercomputer to persistent storage for post-hoc analysis. On exascale systems, the power cost of data movement and the worsening I/O bottleneck will make it necessary for most simulation data to be analyzed in situ, or on the supercomputer while the simulation is running. Furthermore, to meet power consumption constraints, it will be necessary to sharply reduce the volume of data moved on the machine and especially the data that are exported to persistent storage. The combination of sharp data reduction and in situ analysis heighten the importance of capturing data provenance, or the record of what has been done to data, to support validation of results and post-hoc use of the data.

The Data Management effort will address the severe I/O bottleneck and challenges of power-constrained data movement, providing storage system software; in situ workflow support, including data curation and provenance capture; and methods of data collection, reduction, curation, organization and discovery.

It is important to note that the activities presented here target not only the management of data from simulation and subsequent analysis of simulation output, but also the management of experimental and observational data brought onto the exascale supercomputer.

The proposed programs for this technical area are described in the following sections:

DM1: Scalable Storage Software Infrastructure

"Scalable storage software infrastructure" refers to the system software responsible for the reliable storage and retrieval of data supporting and resulting from the execution of applications, check pointing and restart, data capture, and data analysis I/O workloads. Research and advanced development in this area will address the following challenges:

- Scale of storage systems will be so large that current designs cannot be counted on to remain highly available and not lose or damage data.
- New interfaces and capabilities are needed to address bottlenecks and to enable in situ/transit active data manipulation to minimize retained data volume and data movement.

This project aims to provide one or more production grade exascale storage infrastructure(s), from application interfaces to low-level storage organization, to meet requirements for performance and resilience and manage complex exascale storage hierarchies.

DM2: Workflow and Provenance Infrastructure

In situ workflow support and provenance capture will facilitate the execution of complex computational science processes at the exascale facility and the capture and management of information necessary to interpret data over the long term.

Because power constraints and the I/O bottleneck severely limit ability to move data around, store data, and perform offline analysis, scientific applications may have significantly different workflows than today. While workflow systems do not play a major role at the leadership facilities today, the need to analyze data *in situ* and to reduce data movement generally may result in an increased dependence on workflow tools. This project will provide a production grade *in situ* workflow execution system, to be integrated with vendor resource management system (RMS), which will meet science team requirements, including support for user-defined and system-provided provenance capture and retention within the exascale platform.

DM3: Data Collection, Reduction, and Transformation

Data will be so voluminous and data movement so expensive in time and power that applications will have to manipulate data *in situ*, during transit. Thus active storage/ networking techniques are needed in order to extract actionable information from the data. Beyond facilitating the movement of data, integration of data manipulations such as transformation, reduction, or analysis into the data management software stack and data path (including after it is on storage) would enable data to be transformed where it resides or as it is moved. This general capability, termed *system-wide data manipulation*, not only can reduce data movement but also may be used to derive key quantities at runtime to accelerate analysis. Research in this area will focus on developing data models and algorithms/methods for data manipulation, data reduction, and methods of transformation to prepare data for analysis.

This project will provide a production grade system-wide data manipulation infrastructure, including user interface and infrastructure for moving and manipulating data within the system, to be integrated with vendor system services and to meet science team requirements for scalability, functionality, and reliability.

DM4: Data Organization and Discovery

As the amount of data produced by simulations increases, so does the overhead of organizing that data, both in terms of storage space and bandwidth and in terms of scientists' time and effort. Metadata is information about the data, and under this umbrella both storage system metadata (that is, what data is on what devices in the system, how data is referenced) and user metadata (that is, information about organization of stored data, provenance information) are included.

Research and advanced development in this area will include methods of organizing and finding relevant data to enable both *in situ* analysis and post-hoc analysis and reuse of data science at exascale.

This project will provide production grade metadata management to enable application and system metadata capture, indexing of this data, and identification and retrieval of subsets of data based on complex search criteria. It will also ensure that these technologies target science team requirements for scalability, functionality, and reliability.

DM5: Paradigm-Changing Data Management R&D

The Data Management effort relies heavily on the storage industry for solutions that may be adopted to support science at extreme scale. This project provides the research and development for "Plan B" in case industrial solutions are not compatible with the DOE application requirements.

The focus areas for this work will be:

- Data models, formats, and underlying abstractions
- Heterogeneous architectures
- Integration with RMS
- Extreme scale

This project will provide a proof of concept implementations of key building blocks for future science-focused data management systems.

6 Hardware Architecture

The role and responsibility of the Hardware Architecture team is to coordinate and communicate HA Architecture related technical information and inquiries to and from all ExaRD funded activities. This activity organizes inputs from other efforts to identify and prioritize technical challenges and solutions.

The proposed programs for this technical area are described in the following sections:

HA1: DOE-Led Hardware Architecture R&D

This project is envisioned to cover three separate phases focusing on advanced concepts for data movement capabilities.

The Hardware Architecture effort will develop a set of Proxy Architectures that capture advanced concepts for data movement capabilities. These proxy architectures will be open, and enable other research groups and vendors to examine the design methodology in detail. They will be exercised with our simulation infrastructure and proxy applications that come from the Co-Design projects.

In addition, the Hardware Architecture effort will develop and evaluate more effective and more programmable mechanisms for data locality management that may benefit future many-core computing technology. The Hardware Architecture effort will also quantitatively demonstrate the effectiveness of the hardware and software innovations of a data-centric computing paradigm, and independently assess technologies for improving data movement efficiency that emerge from the Fast Forward and Design Forward investments.

Finally, Hardware Architecture will develop a conceptual design for a data movement-centric architecture that could be implemented with the System on a Chip eco-system.

HA2: Design Space Exploration

The HA effort will organize the use of modeling tools, simulation capabilities, advanced architecture testbeds and emulation platforms to enable more credible and comparable results for the co-design process, and strengthen the influence that DOE can have on the decisions of the Fast Forward and Design Forward industry projects. This includes support for existing architectural simulation tools and expertise to apply them to challenging Design Space Exploration questions that arise in the co-design process. These tools are intended to provide an agile rapid prototyping environment to enable us to easily create simplified *proxy architectures* to represent potential realizations of future machine architectures. These *proxy architectures* are intended to provide a quantitative basis for a set of simpler abstract machine models (AMMs) that will give DOE application and algorithm developers a starting point to reason about how their applications map to future machine architectures.

HA3: Pre-production and Prototype node and Rack-scale testbeds

In order to prepare applications, software infrastructure, and our centers for future hardware, it will be valuable to provide early access to rack-scale pre-production compute node hardware for experimentation. This can range from early release production hardware to one-of-a-kind experimental prototypes.

HA4: SoC Design for DOE HPC Applications

The DOE Exascale R&D strategy is based on working with today's leading commodity processor companies. Because these market leaders have unavoidable pressure to protect their existing product lines, it is possible that innovative options to address DOE application requirements will be left unexplored. This project has roots in the DOE R&D investments in semi-custom processor/memory designs that are enabled by the SoC eco-system. It provides contingency plans to develop semi-custom processor designs that can bring embedded processor companies to the HPC market, or provide proof-of-concept hardware prototypes that can lower the "risk barrier" that would prevent adoption by one or more of the mainstream commodity processor companies.

7 System Engineering and Integration

The System Engineering and Integration team will monitor hardware and software projects involving the overall exascale system design to meet the DOE facility requirements and the needs of the DOE mission. Equally important, will be efforts to ensure the integration of computing technologies from across the ExaRD efforts and industry into the overall system design. The team will identify gaps and evaluate system level technologies and research for capabilities that existing market forces would not ensure or for their impact on accelerating industry roadmaps.

The proposed programs for this technical area are described in the following sections:

SI1: Integration of ECI Technologies

One of the critical responsibilities of the System Engineering team is to evaluate the output from across the ExaRD technical areas and integrate the best technologies into an overall system design that meets the DOE facility requirements and the needs of the DOE mission. To accomplish this responsibility, the team will monitor the efforts of other teams through regular communication with the respective leads. The team will also gather the requirements of the DOE HPC facilities in both ASC and Office of Science. As technologies from the various areas mature and are proven out, they will be made available to vendors to incorporate into the overall exascale system design.

SI2: Scalable and Dynamic System Management

The System Engineering and Integration team also monitor system research not covered by the other ExaRD technical areas. One of these areas is the development of scalable and dynamic system management software. System management 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, integration into the site file systems, and integration into the site power, space, and cooling infrastructure.

The research in system management will proceed in three phases. The focus of the first phase is scaling the system management from being able to handle a hundred thousand nodes today to being able to manage a million nodes in an exascale system efficiently and effectively. The second phase goes a step further and assumes that the huge number of nodes is dynamic and at any time some are up, some are down, and some are in between. The third phase builds on the scalable and dynamic system management software developed in the first two phases and extends it to a quality of service paradigm.

DOE facilities currently use system management software that was developed either by industry or within the laboratories. The latter choice is often adopted due to inadequacies in the industry-developed software. Many of the exascale system management issues will require fundamental, basic research, which serves as an ideal topic for academic work.

SI3: System RAS and Real-time Monitoring

Reliability, Availability, and Serviceability (RAS) and the associated system monitoring are extremely critical in the exascale timeframe. As a result, they are the basis of a separate project within System Engineering and Integration. DOE facilities, which struggle to troubleshoot user and system problems on today's petascale systems, view RAS and the associated system monitor as a primary concern for the administration of exascale systems. Exascale RAS software must advance to provide sufficient system availability to support DOE mission critical science. Real-time monitoring of exascale systems must be coupled with the dynamic system management software, with predictive analysis software to assess the availability and service requirements of various capabilities of the system and to adjust the quality of service of different components across the system. The RAS system interacts with most layers of the software stack, which will require coordination with the Software Stack team.

RAS is a fundamental concern for industry partners, who will thus be responsible for a large portion of this project. However, DOE RAS needs outstrip those of most industry customers.

SI4: Industrial Research and System Prototypes

In addition to the system management and RAS research projects identified above, the gap analysis may identify other issues seen only at scales unique to Exascale. Research in areas such as packaging or cooling are best undertaken by industry and may be necessary.

Success of the ECI requires that it produce fully integrated system solutions that inform DOE production procurements. Many issues addressed by the projects under the System Engineering and Integration team can only be assessed on full system prototypes such as P2. At the same time, DOE applications require early access to full system prototypes to ascertain that the complete systems meet DOE mission needs and to prepare production codes for the eventual production system.

8 Milestones

The ExaRD effort must address significant technological uncertainties and a continual need to incorporate research results into the design process. The following table shows the proposed top-level milestones:

FY	Milestone
2016	Conceptual design of eXtreme Bring-up System (XBUS)
2017	Preliminary design of XBUS
2018	Delivery of P0
2019	Delivery of P1 Initial version of XBUS ported to P0
2020	Evaluation of XBUS performance, functionality and correctness Final design of XBUS
2021	Demonstrate Application Readiness
2022	Delivery of P2
2023	Testing and evaluation of XBUS
2023	Enable the delivery of B1
2024	Demonstration of selected applications performing at exascale performance level during early science runs
2025	Final report on evaluation of productivity measures

Table 1