Program Announcement
To DOE National Laboratories

LAB 12-619

AMENDMENT (issued 12/20/2011)

This Program Announcment has been amended to change the Letter of Intent Due Date from December 19, 2011 to December 21, 2011.

Office of Science

Office of Advanced Scientific Computing Research (ASCR)

2012 X-Stack: Programming Challenges, Runtime Systems, and Tools

GENERAL INQUIRIES ABOUT THIS LAB ANNOUNCEMENT SHOULD BE DIRECTED TO:

Technical/Scientific Program Contacts:

Program Manager: Dr. Sonia R. Sachs, (301) 903-0060
Office of Advanced Scientific Computing Research, SC-21.1
E-mail: Sonia.sachs@science.doe.gov

Program Manager: Dr. Lenore Mullin, (301) 903-7113
Office of Advanced Scientific Computing Research, SC-21.1
E-mail: Lenore.mullin@science.doe.gov

SUMMARY:

The Office of Advanced Scientific Computing Research (ASCR) of the Office of Science (SC), U.S. Department of Energy (DOE), hereby invites proposals for basic research that represents significant advances in programming models, languages, compilers, runtime systems and tools that address fundamental challenges related to the system software stack for Exascale computing platforms (X-Stack).

Programming models, languages, and related technologies that have sustained High Performance Computing (HPC) application software development for the past decade are inadequate for Exascale era computers. The significant increase in complexity of Exascale platforms due to energy-constrained, billion-way parallelism, with major changes to processor and memory architecture, requires new energy-efficient and resilient programming techniques that are portable across multiple future machine generations. We expect to make research investments that address fundamental Exascale challenges, while offering a transition path for existing scientific applications to fully explore the challenges and rewards of Exascale platforms.

Exascale programming challenges and strategies were identified in the ASCR Exascale Programming Challenges Workshop [1] and carefully captured in the workshop report [2]. Challenges and strategies related to tools for Exascale platforms were identified in the ASCR Exascale Tools Workshop [3], and captured in the workshop report [4].

Sought are complete solutions that will address multiple components of the system software stack and that will have the following characteristics:

- Scalability: enable applications to strongly scale to Exascale levels of parallelism;
- Programmability: clearly reduce the burden we are placing on high performance programmers;
- Performance Portability: eliminate or significantly minimize requirements for porting to future platforms;
- Resilience: properly manage fault detection and recovery at all components of the software stack; and
- Energy Efficiency: maximally exploit dynamic energy saving opportunities, leveraging the tradeoffs between energy efficiency, resilience, and performance.

We encourage solutions that involve radically new approaches to programming Exascale applications and algorithms. New approaches are required in order to address the complexities of Exascale systems. It is important to demonstrate the viability of such solutions in a broad high performance programming context by showing how the proposed solution:

- interoperates with existing programming environments based on the MPI+X model, so that a smooth migration path is enabled, and/or
- enables the automatic transformation of applications (possibly with users in the loop) from the "old" programming environment to the "new" one, such that the transformations are semantics and performance preserving.

More specific information is included under SUPPLEMENTARY INFORMATION below.

A companion Funding Opportunity Announcement (FOA) DE-FOA-0000619 will also be posted at grants.gov and on the SC Grants and Contracts web site at: http://www.science.doe.gov/grants.

1. Letter of Intent (LOI).

LOI DUE DATE: December 21, 2011

Applicants are strongly encouraged to submit a LOI no later than 11:59 pm December 19, 2011. The LOI should include the following:

1. A cover sheet containing the name, institutional affiliation, e-mail address, and telephone number of the Principal Investigator(s), and Senior/Key personnel expected to be involved in the planned proposal; and the estimated annual cost and total cost of the project over the three-year project period.

2. A 1-2 pages overview of the research plan.

The LOI will be used to organize and expedite the merit review process. Consequently, the submission of a LOI is strongly encouraged but not required. The absence of a LOI will not negatively affect a thorough evaluation of a responsive formal proposal submitted in a timely fashion. The LOI should be sent by E-mail as a PDF file to: ascr-cs@science.doe.gov. Please include the phrase "Letter of Intent" in the subject line.

PROPOSAL DUE DATE:

Formal proposals submitted in response to this Program Announcement must be submitted from the Laboratory to the site office through Searchable FWP by **February 6, 2012, 11:59 p.m. Eastern Time**, to be accepted for merit review and to permit timely consideration for award in Fiscal Year 2012. **Each proposal should be in a single PDF file. The first few pages of the PDF should be the Field Work Proposal followed in the same PDF by the full technical proposal. You are encouraged to transmit your proposal well before the deadline. PROPOSALS RECEIVED AFTER THE DEADLINE WILL NOT BE REVIEWED OR CONSIDERED FOR AWARD.**

SUBMISSION INSTRUCTIONS:

LAB administrators should submit the entire LAB proposal and Field Work Proposal (FWP) via searchable FWP (https://www.osti.gov/fwp). Questions regarding the appropriate LAB administrator or other questions regarding submission procedures can be addressed to the Searchable FWP Support Center. All submission and inquiries about this Program Announcement must reference Program Announcement LAB 12-619

SUPPLEMENTARY INFORMATION:

With the growth in clock frequency stalled, performance advances are now being achieved by an exponential growth in the number of processing elements per chip and growing hardware threading per core. The number of "cores" or explicitly parallel computational elements per chip is likely to double every 18-24 months henceforth. Power has rapidly become the leading design constraint for future HPC systems. New approaches will not emerge from evolutionary changes in processor speed and scale from today's Petascale systems, but will require fundamental breakthroughs in hardware technology, programming models, algorithms, and software at both the system and application level.

As complex memory systems, including 3D memories, are essential components Exascale architectures, a number of questions are raised as a new memory model is created. This essential component of Exascale platforms will impact the design of lightweight mechanisms for memory management, memory virtualization, and for data placement, caching and migration, all of which impact the system software stack.

Exascale systems are expected to have approximately 3-5 orders of magnitude more concurrency than current Petascale platforms. Such systems present both opportunities and challenges to scientific applications and the software stack that supports their ability to express and manage up to a billion separate threads. New algorithms will be required that can exploit vastly more parallelism than existing algorithms without requiring the same order of magnitude more memory, because the available memory will not scale by the same factor. Expressing these new algorithms will require new programming models and programming language constructs that are not available in existing languages.

Energy constraints and resilience challenges add complexity dimensions to programming Exascale systems, so that understanding and leveraging the tradeoffs between energy efficiency, resilience, and performance will be paramount for Exascale systems. Given that minimizing data movement will be critical for energy-constrained architectures, new parallel algorithms with improved locality of reference are required. In addition, active energy and power management may require remapping software dynamically to adjust to changing resource availability, with fine-grained controls in order to maximally exploit dynamic energy saving opportunities. Exascale systems are expected to have a low Mean Time to Interrupt (MTTI) and to suffer from undetected soft hardware errors, leading to a high failure rate of hardware components. These concerns−added to increased component counts, increased software complexity, and numerical accuracy at Exascale−will require radically new approaches to resilience that will involve fault detection and recovery at all components of the software stack.

The complexity of Exascale systems in terms of architectural attributes of concurrency, locality, hierarchy, and heterogeneity is significantly increased from previous machine generations, inhibiting the ability to program such systems. Significant advances in programming models, programming languages, compilers, runtime systems and tools will be needed in order to maximize concurrency, properly deal with asynchrony of computation and communication, exploit data locality, deal with deep memory hierarchies, minimize data movement, hide latencies, manage faults, deal with heterogeneous computing elements, and yet be easily programmable by application developers. Application developers recognize the major disruptions expected in Exascale systems, and are aware that they will need to rewrite their applications. However, application developers need to be assured that the return on their effort can be leveraged for future generations of HPC platforms.

Specific Supplementary Information on Exascale Programming Models and Languages:
Developing high-performance code for an application will involve multiple levels of representations, with semantic and performance preserving transformations that map high-level specifications of a problem into lower level ones, with the lowest level being an executable that is compile and runtime optimized to a particular platform. Semantic and performance preserving transformations enable optimizations to be accomplished without knowledge of how lower layers are implemented, which is fundamental for performance portability.

Each component of this programming stack is associated, at least implicitly, with an abstract machine and a programming model. An abstract machine exposes some but not all features of the platform, and the programming model permits the specification and optimization of how those features are used by the program without having to deal with the complexity of the full machine. Abstract machine models were discussed at the DOE ASCR 2011 Workshops on Architecture I [5], Architecture II [6] and conclusions regarding these models for Exascale platforms are presented at the workshop report [7].

Different kinds of programmers will be involved in developing Exascale applications. Programmers with expertise in the science domain need a more declarative programming style that emphasizes the semantics of the domain, whereas programmers with hardware/software stack expertise can use an imperative programming model that provides full control over mappings to hardware architecture. Functional semantics, which can be incrementally inserted into a language rather than limited to new functional programming languages, will have an important role in novel programming environments.

We expect that high level specifications will use domain specific languages (DSLs) or embedded DSLs to capture the mathematics needed by domain scientists, enabling them to focus on their science rather than the fine details of a complex Exascale system. Automation will be used in the programming stack transformations, significantly reducing the burden placed on high performance programmers, and providing consistency in results, performance, and range of possibilities explored. We also expect that at each component of the programming stack, “reverse mappings” capture execution information of a lower layer component, mapping it to the constructs of a programming model of a higher layer component.

Solutions in Programming Models and Languages may include, but are not limited to, novel strategies in the areas of:

- DSLs and Embedded DSLs that enable domain properties to be used in the optimization of programs at the highest abstraction levels;
- New programming abstractions that virtualize the notion of a core and threading application Programming Interfaces (APIs) with expanded semantics for thread control, placement, launching, and synchronization; New programming abstractions and mechanisms to express memory locality such that data movement through the memory hierarchy is addressed and portability across platforms with different memory hierarchies is guaranteed;
- Novel declarative paradigms to deal with asynchronous computations and fine-grained nested parallelism, while enabling the joint optimization among techniques for algorithm exploration, representation exploration, parallelization, placement, and scheduling; and
- Automated techniques that transform domain-specific abstract representations of computations into multiple intermediate abstract representations on the path to a runtime optimized code, as elaborated in section 4.1 of [2].

Specific Supplementary Information on Exascale Compilers and Runtime Systems:
A paradigm shift is required to achieve Exascale computing, as following conventional practice may undermine our goal of high performance, low power Exascale computations. This shift is represented by the execution model [8], which is responsible for orchestrating all aspects of executions on a particular machine. The execution model for a computer determines the design of associated abstract machine models, compilers, and runtime systems. Current research investment on this area is ongoing [9], [10], and is expected to inform future investments in compiler and runtime technologies, including awards that result from this Program Announcement.

Compilers will be the natural place to implement optimizations that explore options for discretization, data representation, scheduling, placement, and choice of solvers. Coupled with hierarchical processing and memory structures, compiler management of parallelism, data locality and data movement across the system will become even more important to performance and also essential to managing power and energy. Driven by the complexity of compiler mapping and optimization technology, interfaces among compiler, programming model and run-time system will need to be redesigned.

The biggest disruption in the path to Exascale will occur at the intra-node level, due to severe memory and power constraints per core, 3X increase in the degree of intra-node parallelism, and to the vast degrees of performance and functional heterogeneity across cores. These challenges clearly point to radically new approaches to intra-node runtime systems. Functions expected from novel, self-aware, resilient runtime systems are the autonomic management of resources, dynamic load balancing, latency hiding mechanisms, management of data movement and locality, active power management, and detection and recovery from faults.

Advanced runtime systems that support new programming models and languages are needed. New runtime approaches can be used to rethink automatic parallelization of applications, with results that have not been possible up to date with compile-time approaches. Novel runtime mechanisms include fast synchronization of operations, as well as lightweight and adaptive communication configuration and management that enable the efficient mapping of the communication graph onto the underlying hardware interconnection topology. Schedulers must dynamically maximize resource utilization and minimize work starvation and resource contention, all while avoiding deadlocks and dealing with powered off resources and hardware features. Efficient locality-sensitive scheduling of the billion-way tasks, including task placement and migration, is a major research challenge to be addressed. Fault handling through transparent task migration and system reconfiguration adds yet another layer of complexity to this challenge. Simulation modeling will certainly be helpful for innovative designs in this area.

Solutions in Compilers and Runtime Systems may include, but are not limited to, novel strategies in the areas of:

- Compiler and runtime methods to support fine-grained dynamic parallelism, data locality, heterogeneity, resiliency, and energy efficiency across the system, exploring and optimizing options for discretization, data representation, scheduling, placement and integrating parallelism within and between nodes; Novel compiler and run-time interfaces that are conducive to dynamic behavior and empirical search-based optimization techniques. Compiler support that automates transformations of code from their semantic description to their implementation on a specific architecture;
- Compiler support for embedded DSLs and for transformations of code from their high level semantic description to their implementation on a specific architecture; Novel runtime approaches to enable auto-parallelization of applications; and
- Self-aware runtime systems and lightweight OS kernels for the support of efficient and dynamic communications, synchronization, scheduling, task placement and migration, as well as the autonomic management of resources, identifying and reacting to load imbalances and the intermittent loss of resources.

Specific Supplementary Information on Exascale Tools:

Harnessing the potential of Exascale platforms is a daunting task because of the unprecedented complexity of these systems. Applications, software stack, and tools face similar challenges at Exascale and will need to concurrently evolve.

Advanced tools should be co-designed with programming models, runtime systems, operating systems, and hardware architectures, so that tool interfaces and requirements are integrated into the stack components. These tools need to support the multiple levels of abstractions that will be required in Exascale applications, from domain-specific, to intermediate representations, to low-level abstractions, focusing on several different audiences and providing the necessary information for high-level users, who rely only on the provided abstractions, and system users, who have a solid understanding of the system complexities and are willing to break abstractions where necessary to achieve performance.

Exascale programmers will need a new generation of performance tools that help users assess how efficiently the billion-way concurrency is being employed, how well applications are dynamically adapting to faults and varying hardware performance, how well applications are taking advantage of the available memory hierarchy, how much unnecessary data replication is present, what data movement and energy efficiency opportunities are left unexplored, how efficient are resources allocated, what is the impact of contention for shared resources–all of which is correlated to the application source code and provides insights and automated methods to prevent or mitigate performance problems.

Advanced performance tools will need to leverage hardware monitoring capabilities to identify inefficient access patterns, quantify the costs of these inefficiencies, and provide guidance as to how the code can be improved. These tools should be able to monitor health and status of system resources, including fault detection, mapping captured information into the software stack, which in turn is used in the optimization of application codes while they run (e.g., process migration may be triggered by the identification of load imbalance by the performance analysis tool).

In order to deal with the Exascale levels of concurrency, tools will have to manage a flood of data and, as a consequence, comprehensive execution tracing to a central storage location for post mortem analysis will be infeasible for the full system. To measure long-running executions in their entirety, only tools that record compact execution profiles will be practical.

Exascale also demands a new generation of debugging tools that automatically or semi-automatically reduce the problem to smaller core counts. Tool support for debugging at Exascale should range from simple approaches that cluster processes into similar groups, to automatic root cause analysis that directly points users to the most probable causes for observed behaviors. Debugging solutions should be capable of combining static information extracted from an application's source or binary code with dynamically gathered and aggregated data.

Debugging and correctness tools are expected to operate at the full scale of the Exascale platform, deal with heterogeneous hardware, specialized memory systems, and with hardware, system software and applications that are highly adaptive to changing system conditions. Debugging at a large scale should include problem reduction methods (e.g., group operations), automatic analysis techniques (e.g., outlier detection, various forms of clustering, automatic model generation), and root cause detection techniques. The need to virtualize new hardware support for managing and accessing memory efficiently will likely require automated transformations, which in turn require mechanisms to verify the correctness of such transformations in order to guarantee an equivalent execution. Debugging tools should offer interfaces that expose hardware features and how code is executed on the underlying hardware and associated software stack. Introspection capabilities (such as memory reference tracing, external environment control, etc.) will be key to the effectiveness of tools and interacting components.

Beyond performance analysis and debugging tools, a new category of tools will be needed in Exascale systems such that developing code “correct by design” is enforced. Tools for correctness include techniques for preventing problems in the code as part of its development, guiding the development of correct code and validating numerous properties via proof techniques. Model checking tools, narrowed to address high performance computing requirements, can be extremely useful to support application development because they not only detect problems in a very comprehensive way (100% of state space coverage), but also provide examples of how the problems were caused, having a much greater impact than a list of issues that could be mostly false positives. Left untamed, lack of correctness of scientific code will have catastrophic consequences in the Exascale era.

Another new category of advanced tools involves parallelization of code and code refactoring/transformation, including acceleration discovery. Refactoring tools will be needed to automatically adapt applications to Exascale environments. Acceleration tools will automatically or semi-automatically, at compile time or at runtime, identify code regions that are suitable for acceleration, outlining them into separate code pieces and transforming them into specialized code for the accelerator hardware.

Given that different tools often share needs (e.g. code browsing, or binary analysis), support for tool components and mechanisms for sharing tool infrastructure will be critical to effectively develop the tools required for Exascale, lowering development costs and delivering improved usability to users.

Exascale Tools Solutions may include, but are not limited to, novel strategies in the areas of:

- Automatic analysis capabilities to measure and analyze thread metrics, concurrency and locality metrics, data movement and energy efficiency, resource utilization and contention, bottlenecks and root causes, correlating results to application representations, at the various levels of the stack;
- Support for new programming models and runtime system, in a closed feedback loop with the hardware architecture, providing insight about elements of the execution environment;
- Novel correctness methods that enable/enforce “correct-by-design” code; Novel debugging methods that identify and mitigate errors, automatically or semi-automatically reducing the problem to some form of hierarchical debugging; and
- Code parallelization, refactoring and transformation methods needed to automatically adapt applications to Exascale environments, including acceleration tools that identify and transform code regions suitable for acceleration into specialized hardware code.

References

[1] DOE ASCR 2011 Exascale Programming Challenges Workshop: http://science.energy.gov/ascr/research/computer-science/programming-challenges-workshop/

[2] DOE ASCR 2011 Exascale Programming Challenges Workshop Report: http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/ProgrammingChallengesWorkshopReport.pdf

[3]DOE ASCR 2011 Exascale Tools Workshop: http://science.energy.gov/ascr/research/computer-science/exascale-tools-workshop/

[4] DOE ASCR 2011 Exascale Tools Workshop Report: http://science.energy.gov/ascr/news-and-resources/program-documents/docs/ExascaleToolsWorkshopReport.pdf

[5] DOE ASCR 2011 Workshop on Architectures I: Exascale and Beyond: Gaps in Research, Gaps in our Thinking, http://www.orau.gov/archI2011/default.htm

[6] DOE ASCR 2011 Workshop on Architectures II: Exascale and Beyond: Configuring, Reasoning, and Scaling, http://www.orau.gov/archII2011/default.htm

[7] Report of the 2011 Workshop on Architectures II: Exascale, and Beyond: Configuring, Reasoning, and Scaling, http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/ArchitecturesIIWorkshopReport.pdf

[8] Vivek Sarkar et al., "Exascale Software Study: Software Challenges in Extreme Scale Systems," DARPA IPTO, September 14, 2009. URL: http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/ECSS%20report%20101909.pdf

[9] Curtis Janssen, Adolfy Hoisie, Thomas Sterling, and John Shalf: "Evaluation, Optimization, and Application of Execution Models for Exascale Computing," http://sites.google.com/site/executionmodelsforexascale/

[10] Bob Lucas, Thomas Sterling, and John Shalf, "Quantifying Overhead in Today's Execution Models" project, https://sites.google.com/site/executionmodelsisilbnl/?pli=1

[11] Mantevo mini-apps, https://software.sandia.gov/mantevo/

[12] C. L. Janssen, H. Adalsteinsson, S. Cranford, J. P. Kenny, A. Pinar, D. A. Evensky, and J. Mayo, "A simulator for large-scale parallel computer architectures", International Journal of Distributed Systems and Technology (IJDST), 1, pp. 57-73, 2010.

[13] Erich Strohmaier et al., "TORCH Computational Reference Kernels: A Testbed for Computer Science Research, University of California, Berkeley, Technical Report, http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-144.html

<u>Additional Proposal Requirements</u>: We are looking for strong teams that address multiple components of the software stack. Collaborative proposals must carefully consider the fact that we will give priority to applications that have a lean budget, in which overheads are minimized and in which every senior/key personnel has a significant technical contribution to the proposed research.

Each proposal must include the following:

1. Description of plans for developing prototypes of the proposed solution;
2. Description of the proposed path to integration and/or interoperation with existing programming environments, including a proposed timeline;
3. Evaluation plan with respect to scalability, programmability, energy efficiency, and performance metrics using compact applications, mini-applications [11], [12] and/or application skeletons [13].

For official postings see the Office of Science Grants and Contracts web site, http://www.science.doe.gov/grants.

<u>Collaborations</u>: Collaborative research projects with other institutions, such as universities, industry, non- profit organizations, and Federally Funded Research and Development Centers (FFRDCs), including the DOE National Laboratories, are strongly encouraged. Collaborative proposals submitted from different institutions should clearly indicate they are part of a proposed collaboration and contain the same title, Abstract and Narrative for that research project. In addition, such proposals must describe the work and the associated budget for the research effort being performed under the leadership of the Principal Investigator at that participating institution. These collaborative proposals should all have the same title as the Lead Institution.

<u>Program Funding</u>: Awards are expected to be made for a period of three years at a funding level of up to $15,000,000 per year to support multiple awards in Fiscal Year 2012, with out-year support contingent on the availability of appropriated funds and satisfactory progress.

DOE is under no obligation to pay for any costs associated with the preparation or submission of a proposal. DOE reserves the right to fund, in whole or in part, any, all, or none of the proposals submitted in response to this Program Announcement.

The instructions and format described below should be followed. You must reference Program Announcement LAB 12-619 on all submissions and inquiries about this program.

OFFICE OF SCIENCE
GUIDE FOR PREPARATION OF SCIENTIFIC/TECHNICAL PROPOSALS TO BE SUBMITTED BY NATIONAL LABORATORIES

Proposals from DOE National Laboratories submitted to the Office of Science (SC) as a result of this Program Announcement will follow the Department of Energy Field Work Proposal process with additional information requested to allow for scientific/technical merit review. The following guidelines for content and format are intended to facilitate an understanding of the requirements necessary for SC to conduct a merit review of a proposal. Please follow the guidelines carefully, as deviations could be cause for declination of a proposal without merit review.

1. Evaluation Criteria

Proposals will be subjected to scientific merit review (peer review) and will be evaluated against the following evaluation criteria which are listed in descending order of importance. Included within each criterion are specific questions that the merit reviewers will be asked to consider:

1. **Scientific and/or Technical Merit of the Project**
    a. Does the proposed research significantly advance the state-of-the-art in programming models, languages, compiler, runtime systems, and tools?
    b. Does the proposed research provide for complete solutions that will address multiple components of the system software stack?
    c. Does the propose research clearly addresses scalability, programmability, performance portability, resilience, and energy efficiency?
    d. Does the proposed research significantly lower the barriers to effectively program Exascale machines?
    e. What is the likelihood that the applicant can overcome the key challenges and, as warranted, shift research directions in response to promising advances in basic research?

2. **Appropriateness of the Proposed Method or Approach**
    a. How well does the research plan address interfaces of the multiple components of the proposed solution?
    b. Does the research plan contain the development of prototypes of the proposed solution?
    c. Does the research plan include demonstration of viability of the proposed solution to interoperate with existing programming environment based on MPI+X, and/or to automatically transform from existing codes to new ones?
    d. Does the research plan include validation strategies using compact, mini-apps, or skeletons of DOE scientific applications?
    e. Does the research plan contain appropriate performance metrics that will allow progress and contributions to be measured?
    f. If this is a collaborative application, does it include a management plan that addresses the organization, communications, and coordination of the collaborating teams?

3. **Competency of Applicant's Personnel and Adequacy of Proposed Resources**
    a. Do the applicants have a proven record of success in delivering results for advanced computational science research?
    b. Do the applicants have a proven record of research and development in the disciplines needed for success in projects that involve integration of multiple software stack components?
    c. Are the roles and intellectual contributions of the Principal Investigator(s), and each senior/key personnel adequately described? Do you consider the contributions of each senior/key personnel of significant value for the project?

4. **Reasonableness and Appropriateness of the Proposed Budget**
    a. Is the applicant's requested budget appropriate? Is the budget as lean as it can be to deliver the promised results? Are the budget overheads minimized?
    b. Does the requested budget support the applicant's specified management structure in a meaningful way? Does the applicant have a process for reallocating funds to address changing priorities?
    c. Does the applicant have a process for reallocating individuals funds to address changing priorities?
    d. Is travel budget appropriate? Are video conferencing technologies proposed to reduce the travel budget?

The selection official will also consider the following program policy and management factors in the selection process:

a. Potential impact of proposed research activities on ASCR Exascale goals in the areas of this Announcement.
b. Potential for developing synergies and/or relation of the proposed research activities to other research efforts supported by ASCR, particularly co-design;
c. Total amount of DOE funds available; and
d. A management plan that addresses the organization, communications, and coordination of the collaborating researchers. This plan should include mitigation strategies for foreseeable risks and explain how the project will have sufficient flexibility to adapt to changing priorities, challenges, and resources.

The evaluation process will include program policy factors such as the relevance of the proposed research to the terms of the Program Announcement and the agency's programmatic needs. Note that external peer reviewers are selected with regard to both their scientific expertise and the absence of conflict-of-interest issues. Both Federal and non-Federal reviewers may be used, and submission of a proposal constitutes agreement that this is acceptable to the investigator(s) and the submitting institution.

2. Summary of Proposal Contents

- Field Work Proposal (FWP) Format (Reference DOE Order 412.1A) (DOE ONLY)
- Proposal Cover Page
- Table of Contents

- Budget (DOE Form 4620.1) and Budget Explanation
- Abstract (one page)
- Narrative (main technical portion of the proposal, including background/introduction, proposed research and methods, timetable of activities, and responsibilities of key project personnel – 25-page limit
- Literature Cited
- Biographical Sketch(es)
- Description of Facilities and Resources
- Other Support of Investigator(s)
- Appendix (optional)

2.1 Submission Instructions

LAB administrators should submit the entire LAB proposal and Field Work Proposal (FWP) via searchable FWP (https://www.osti.gov/fwp). Questions regarding the appropriate LAB administrator or other questions regarding submission procedures can be addressed to the Searchable FWP Support Center. All submission and inquiries about this Program Announcement must reference Program Announcement to DOE National Laboratories LAB 12-619. Full proposals submitted in response to this Program Announcement must be submitted to the searchable FWP database no later than 11:59 pm, Eastern Time, **February 6, 2012**. It is important that the entire peer reviewable proposal be submitted to the searchable FWP system as a single PDF file attachment.

3. Detailed Contents of the Proposal

Adherence to type size and line spacing requirements is necessary for several reasons. No researcher should have the advantage, or by using small type, of providing more text in his or her proposal. Small type may also make it difficult for reviewers to read the proposal. Proposals must have 1-inch margins at the top, bottom, and on each side. Type sizes must be at least 11 point. Line spacing is at the discretion of the researcher but there must be no more than 6 lines per vertical inch of text. Pages should be standard 8 1/2" x 11" (or metric A4, i.e., 210 mm x 297 mm).

3.1 Field Work Proposal Format (Reference DOE Order 412.1A) (DOE ONLY)

The Field Work Proposal (FWP) is to be prepared and submitted consistent with policies of the investigator's laboratory and the local DOE Operations Office. Additional information is also requested to allow for scientific/technical merit review.

3.2 Proposal Cover Page

The following proposal cover page information may be placed on plain paper. No form is required.

Title of proposed project:
SC Program Announcement title and number: **2012 X-Stack: Programming Challenges, Runtime Systems, and Tools - LAB 12-619**

Name of laboratory:
Name of principal investigator (PI):
Position title of PI:
Mailing address of PI:
Telephone of PI:
Fax number of PI:
Electronic mail address of PI:
Name of official signing for laboratory*:
Title of official:
Fax number of official:
Telephone of official:
Electronic mail address of official:
Requested funding for each year; total request:
Use of human subjects in proposed project:

If activities involving human subjects are not planned at any time during the proposed project period, state "No"; otherwise state "Yes", provide the IRB Approval date and Assurance of Compliance Number and include all necessary information with the proposal should human subjects be involved.

Use of vertebrate animals in proposed project:

If activities involving vertebrate animals are not planned at any time during this project, state "No"; otherwise state "Yes" and provide the IACUC Approval date and Animal Welfare Assurance number from NIH and include all necessary information with the proposal.

Signature of PI, date of signature:
Signature of official, date of signature*:

* The signature certifies that personnel and facilities are available as stated in the proposal, if the project is funded.

3.3 Table of Contents

Provide the initial page number for each of the sections of the proposal. Number pages consecutively at the bottom of each page throughout the proposal. Start each major section at the top of a new page. Do not use unnumbered pages, and do not use suffices, such as 5a, 5b.

3.4 Budget and Budget Explanation

A detailed budget is required for the entire project period and for each fiscal year. It is preferred that DOE's budget page, Form 4620.1 be used for providing budget information*. Modifications of categories are permissible to comply with institutional practices, for example with regard to overhead costs.

A written justification of each budget item is to follow the budget pages. For personnel this should take the form of a one-sentence statement of the role of the person in the project. Provide a detailed justification of the need for each item of permanent equipment. Explain each of the other direct costs in sufficient detail for reviewers to be able to judge the appropriateness of the amount requested.

Further instructions regarding the budget are given in section 4 of this guide.

* Form 4620.1 is available at web site: http://www.science.doe.gov/grants/budgetform.pdf

3.5 Abstract

Summarize the proposal in one page. Give the project objectives (in broad scientific terms), the approach to be used, and what the research is intended to accomplish. State the hypotheses to be tested (if any). At the top of the abstract give the lead DOE National Laboratory, project title, names of all the investigators and their institutions, and contact information for the principal investigator, including e-mail address.

3.6 Narrative (main technical portion of the proposal, including background/introduction, proposed research and methods, timetable of activities, and responsibilities of key project personnel).

The narrative comprises the research plan for the project and is limited to a **maximum of 25 pages**. It should contain enough background material in the Introduction, including review of the relevant literature, to demonstrate sufficient knowledge of the state of the science. The major part of the narrative should be devoted to a description and justification of the proposed project, including details of the methods to be used. It should also include a timeline for the major activities of the proposed project, and should indicate which project personnel will be responsible for which activities. It is important that the 25-page technical information section provide a complete description of the proposed work, because reviewers are not obliged to read the Appendices. Proposals exceeding these page limits may be rejected without review or the first 25 pages may be reviewed without regard to the remainder.

The page count of 25 does not include the Cover Page and Budget Pages, the Title Page, the biographical material and publication information, or any Appendices. However, it is important that the 25-page technical information section provide a complete description of the proposed work, since reviewers are not obliged to read the Appendices. Please do not submit general letters of support as these are not used in making funding decisions and can interfere with the selection of peer reviewers.

Background

Background – explanation of the importance and relevance of the proposed work.

Proposed Research and Tasks

In addition to the technical description of the proposed work and tasks, include a discussion of schedule, milestones, and deliverables.

Is this a Collaboration? If yes, please list ALL Collaborating Institutions/PIs* and indicate which ones will also be submitting proposals. Also indicate the PI who will be the point of contact and coordinator for the combined research activity.

* Note that collaborating proposals must be submitted separately. However, if you are submitting as a Lead Institution, in addition to meeting all criteria for submitting a peer reviewable proposal, please provide the following information in the form of a table as shown below:

Sample Table for the Lead Institution ($ in thousands)

2012 X-Stack	Year 1	Year 2	Year 3	Year 4	Year 5	Total
Name of the Principal Investigator and Institution	$	$	$	$	$	$
Collaborating Institutions						**Total**
Name of Co-PI and Institution	$	$	$	$	$	$
Name of Co-PI and Institution	$	$	$	$	$	$
Name of Co-PI and Institution	$	$	$	$	$	$
TOTALS	$	$	$	$	$	$

3.7 Literature Cited

Give full bibliographic entries for each publication cited in the narrative. Each reference must include the names of all authors (in the same sequence in which they appear in the publication), the article and journal title, book title, volume number, page numbers, and year of publication. Include only bibliographic citations. Principal investigators should be especially careful to follow scholarly practices in providing citations for source materials relied upon when preparing any section of the proposal.

3.8 Biographical Sketches

This information is required for senior personnel at the institution submitting the proposal and at all subcontracting institutions (if any). The biographical sketch is limited to a maximum of two pages for each investigator and must include:

Education and Training. Undergraduate, graduate and postdoctoral training, provide institution, major/area, degree and year.

Research and Professional Experience. Beginning with the current position list, in chronological order, professional/academic positions with a brief description.

Publications. Provide a list of up to 10 publications most closely related to the proposed project. For each publication, identify the names of all authors (in the same sequence in which they appear in the publication), the article title, book or journal title, volume number, page numbers, year of publication, and website address if available electronically. Patents, copyrights and software systems developed may be provided in addition to or substituted for publications.

Synergistic Activities. List no more than five professional and scholarly activities related to the effort proposed.

To assist in the identification of potential conflicts of interest or bias in the selection of reviewers, the following information must also be provided in each biographical sketch.

> *Collaborators and Co-editors:* A list of all persons in alphabetical order (including their current organizational affiliations) who are currently, or who have been, collaborators or co-authors with the investigator on a research project, book or book article, report, abstract, or paper during the 48 months preceding the submission of the proposal. For publications or collaborations with more than 10 authors or participants, only list those individuals in the core group with whom the Principal Investigator interacted on a regular basis while the research was being done. Also, include those individuals who are currently or have been co-editors of a special issue of a journal, compendium, or conference proceedings during the 24 months preceding the submission of the proposal. Finally, list any individuals who are not listed in the previous categories with whom you are discussing future collaborations. If there are no collaborators or co-editors to report, this should be so indicated.
>
> *Graduate and Postdoctoral Advisors and Advisees:* A list of the names of the individual's own graduate advisor(s) and principal postdoctoral sponsor(s), and their current organizational affiliations. A list of the names of the individual's graduate students and postdoctoral associates during the past five years, and their current organizational affiliations.

3.9 Description of Facilities and Resources

Facilities to be used for the conduct of the proposed research should be briefly described. Indicate the pertinent capabilities of the institution, including support facilities (such as machine shops), that will be used during the project. List the most important equipment items already available for the project and their pertinent capabilities. Include this information for each subcontracting institution (if any).

3.10 Other Support of Investigators

Other support is defined as all financial resources, whether Federal, non-Federal, commercial, or institutional, available in direct support of an individual's research endeavors. Information on active and pending other support is required for all senior personnel, including investigators at collaborating institutions to be funded by a subcontract. For each item of other support, give the organization or agency, inclusive dates of the project or proposed project, annual funding, and level of effort (months per year or percentage of the year) devoted to the project.

3.11 Appendix

Information not easily accessible to a reviewer may be included in an appendix, but **do not use the appendix to circumvent the page limitations of the proposal.** Reviewers are not required to consider information in an appendix, and reviewers may not have time to read extensive appendix materials with the same care they would use with the proposal proper.

The appendix may contain the following items: up to five publications, manuscripts accepted for publication, abstracts, patents, or other printed materials directly relevant to this project, but not generally available to the scientific community. If letters of endorsement are included in a proposal, they will be removed before the proposal is submitted for review.

4. Detailed Instructions for the Budget (DOE Form 4620.1 "Budget Page" may be used).

4.1 Salaries and Wages

List the names of the principal investigator and other key personnel and the estimated number of person-months for which DOE funding is requested. Proposers should list the number of postdoctoral associates and other professional positions included in the proposal and indicate the number of full-time-equivalent (FTE) person-months and rate of pay (hourly, monthly or annually). For graduate and undergraduate students and all other personnel categories such as secretarial, clerical, technical, etc., show the total number of people needed in each job title and total salaries needed. Salaries requested must be consistent with the institution's regular practices. The budget explanation should define concisely the role of each position in the overall project.

4.2 Equipment

DOE defines equipment as "an item of tangible personal property that has a useful life of more than two years and an acquisition cost of $50,000 or more." Special purpose equipment means equipment which is used only for research, scientific or other technical activities. Items of needed equipment should be individually listed by description and estimated cost, including tax, and adequately justified. Allowable items ordinarily will be limited to scientific equipment that is not already available for the conduct of the work. General purpose office equipment normally will not be considered eligible for support.

4.3 Domestic Travel

The type and extent of travel and its relation to the research should be specified. Funds may be requested for attendance at meetings and conferences, other travel associated with the work and subsistence. In order to qualify for support, attendance at meetings or conferences must enhance the investigator's capability to perform the research, plan extensions of it, or disseminate its results. Consultant's travel costs also may be requested.

4.4 Foreign Travel

Foreign travel is any travel outside Canada and the United States and its territories and possessions. Foreign travel may be approved only if it is directly related to project objectives.

4.5 Other Direct Costs

The budget should itemize other anticipated direct costs not included under the headings above, including materials and supplies, publication costs, computer services, and consultant services (which are discussed below). Other examples are: aircraft rental, space rental at research establishments away from the institution, minor building alterations, service charges, and fabrication of equipment or systems not available off- the-shelf. Reference books and periodicals may be charged to the project only if they are specifically related to the research.

a. Materials and Supplies

The budget should indicate in general terms the type of required expendable materials and supplies with their estimated costs. The breakdown should be more detailed when the cost is substantial.

b. Publication Costs/Page Charges

The budget may request funds for the costs of preparing and publishing the results of research, including costs of reports, reprints page charges, or other journal costs (except costs for prior or early publication), and necessary illustrations.

c. Consultant Services

Anticipated consultant services should be justified and information furnished on each individual's expertise, primary organizational affiliation, daily compensation rate and number of days expected service. Consultant's travel costs should be listed separately under travel in the budget.

d. Computer Services

The cost of computer services, including computer-based retrieval of scientific and technical information, may be requested. A justification based on the established computer service rates should be included.

e. Subcontracts

Subcontracts should be listed so that they can be properly evaluated. There should be an anticipated cost and an explanation of that cost for each subcontract. The total amount of each subcontract should also appear as a budget item.

4.6 Indirect Costs

Explain the basis for each overhead and indirect cost. Include the current rates.