|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. WORK PROPOSAL NO. ERKJ247 | | | | | 2. REVISION NO.  000 | | 3. DATE PREPARED  03/29/2013 | | | | | | 94 | | |
| 4. WORK PROPOSAL TITLE: MCREX: Using Monte Carlo Algorithums to Achieve Resiliency and Performance | | | | | | | | | | | 5. BUDGET AND REPORTING CODE  KJ0402000 | | | | |
| 6. WORK PROPOSAL TERM  BEGIN: 10/01/2012 END: 09/30/2015 | | | | | | PATENT STATUS  This proposal is being transmitted in advance of patent review for evaluation purposes only. No further dissemination or publication shall be made without prior approval of the Assistant General Counsel for Patents, DOE. | | | | | | 7. Is This Work Proposal Included in the Institutional Plan?  Yes No | | | |
| NAME: (Last, First, MI) (Phone Number)  8. HEADQUARTERS/OPERATIONS OFFICE PROGRAM MANAGER: Pao, Karen (301) 903-5384 | | | | | | 11. HEADQUARTERS ORGANIZATIONS: **Science** | | | | | | 14. DOE ORGANIZATION CODE:  **SC** | | | |
| 9. OPERATIONS OFFICE WORK PROPOSAL REVIEWER: Lin, Wayne C (865)576-0639 | | | | | | 12. FIELD OFFICE:  **Oak Ridge Operations** | | | | | | 15. DOE ORGANIZATION CODE:  **ON** | | | |
| 10. CONTRACTOR WORK PROPOSAL PRINCIPAL INVESTIGATOR(S)/MANAGER: Evans, Thomas M (865)576-3535 | | | | | | 13. CONTRACTOR NAME: **Oak Ridge National Laboratory Managed by UT-Battelle, LLC For the U.S. Department of Energy Post Office Box 2008 Oak Ridge, TN 37831** | | | | | | 16. DOE CONTRACTOR CODE:  **41** | | | |
| 17. WORK PROPOSAL DESCRIPTION (Approach, anticipated benefits in 200 words or less)  The next generation of computational science applications will require numerical solvers that are capable of high performance on proposed HPC platforms. In order to meet this goal, solvers must be resilient to soft and hard system failures, provide high concurrency on heterogeneous hardware configurations, and retain numerical accuracy and efficiency. In light of these requirements, a natural avenue of inquiry would be to adapt the current stable of numerically efficient solvers to this new high-performance computing regime. However, an alternative approach would be to investigate different classes of algorithms that can address issues of resiliency, particularly fault tolerance and hard processor failures, naturally. In this proposal, the team will investigate new stochastic methods for solving linear systems, otherwise termed Monte Carlo Resilient, Exascale (MCREX) solvers. The family of methods that the team has proposed builds on the sequential Monte Carlo work of Halton, 1962. While showing significant promise, this class of solvers has not made inroads into the broader computational science community. The methods that the team has initially developed use Monte Carlo to accelerate a fixed-point iteration; therefore, the team has called them Monte Carlo Synthetic Acceleration (MCSA). Preliminary work using MCSA has demonstrated that they are at least as efficient as Jacobi-preconditioned Conjugate Gradient (PCG) on sparse, SPD systems. These initial results demonstrate that very good efficiency could be attained on non-symmetric systems; thus making MCSA an ideal solver in non-linear Newton schemes. Furthermore, Monte Carlo methods have the benefit of addressing resiliency in a natural way; soft errors can be treated as high variance samples and lost histories from processor failures can be easily discarded without affecting the quality of the solution. | | | | | | | | | | | | | | | |
| 18. CONTRACTOR WORK PROPOSAL MANAGER: (Name and Phone No.)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Signature) Nichols, Jeff ALD (865)574-6224 (Date) | | | | | | | | | | 19. OPERATIONS OFFICE REVIEW OFFICIAL  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Signature) (Date) | | | | | |
| 20. DETAIL ATTACHMENTS:  (See instructions for page 3) | | | | | | | | | | | | | | | |
|  | a. Facility Requirements | X | e. Approach | | | |  | i. NEPA Requirements | | | | |  | m. ES&H Considerations |
|  | b. Publications |  | f. Technical Progress | | | | X | j. Milestones | | | | |  | n. Human/Animal Subjects |
| X | c. Purpose |  | g. Future Accomplishments | | | |  | k. Deliverables | | | | | X | o. Other (Specify) |
| X | d. Background |  | h. Relationships To Other Projects | | | |  | l. Perform Measures/Expectations | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21. STAFFING (in staff years) | FY 2013 | FY 2014 | FY 2015 | | | FY 2016 | FY 2017 | | TOTAL TO COMPLETE |
| REQUEST | | AUTHOR. |
| a. SCIENTIFIC / OTHER DIRECT - ORNL | 0.9 | 0.9 | 0.8 | |  | <!STAFFING\_SCI\_FY4!> | <!STAFFING\_SCI\_FY5!> | | <!STAFFING\_SCI\_FYT!> |
| b. OTHER DIRECT - OTHER SITES |  |  |  | |  | <!STAFFING\_OTH\_FY4!> | <!STAFFING\_OTH\_FY5!> | | <!STAFFING\_OTH\_FYT!> |
| c. TOTAL DIRECT | 0.9 | 0.9 | 0.8 | |  | <!STAFFING\_TOT\_FY4!> | <!STAFFING\_TOT\_FY5!> | | <!STAFFING\_TOT\_FYT!> |
| 22. OPERATING EXPENSE (in Thousands) |  |  |  | |  |  |  | |  |
| a. TOTAL OBLIGATIONS | 300 | 300 | 300 | |  | <!OPER\_OBS\_TOT\_FY4!> | <!OPER\_OBS\_TOT\_FY5!> | | <!OPER\_OBS\_TOT\_FYT!> |
| COSTS: |  |  |  | |  |  |  | |  |
| 1) WAGE POOL AND ORG. BURDEN | 215 | 216 | 217 | |  | <!OPER\_COSTS\_WP\_FY4!> | <!OPER\_COSTS\_WP\_FY5!> | | <!OPER\_COSTS\_WP\_FYT!> |
| 2) MATERIALS AND SERVICES | 8 | 7 | 7 | |  |  |  | |  |
| 3) SUBCONTRACTS AND CONSULTANTS | 0 | 0 | 0 | |  | <!OPER\_COSTS\_SUB\_FY4!> | <!OPER\_COSTS\_SUB\_FY5!> | | <!OPER\_COSTS\_SUB\_FYT!> |
| 4) INDIRECT COSTS | 77 | 77 | 76 | |  |  |  | |  |
| b. TOTAL COSTS | 300 | 300 | 300 | |  | <!OPER\_COSTS\_TOT\_FY4!> | <!OPER\_COSTS\_TOT\_FY5!> | | <!OPER\_COSTS\_TOT\_FYT!> |
| 23. EQUIPMENT (in Thousands) |  |  |  | |  |  |  | |  |
| a. EQUIPMENT OBLIGATIONS |  |  |  | |  |  |  | |  |
| b. EQUIPMENT COSTS |  |  |  | |  |  |  | |  |
| 24. MILESTONE SCHEDULE (TASKS:) | | DOLLARS (in Thousands) | | | | SCHEDULE (DATE) | | | |
| PROPOSED | | AUTHORIZED | | PROPOSED | | AUTHORIZED | |
| Demonstrate convergence properties and performance of MCSA algorithm on sparse symmetric and non-symmetric systems. | |  | |  | | 09/13 | | 00/00 | |
| Show resiliency of MCSA algorithm for soft and hard errors while solving the linear advection-diffusion-reaction (ADR) model problem. | |  | |  | | 09/14 | | 00/00 | |
| Show parallel performance of MCSA algorithm on existing HPC architectures and demonstrate scaling to exascale systems using the MCSA performance model. | |  | |  | | 09/15 | | 00/00 | |
|  | |  | |  | |  | |  | |
|  | |  | |  | |  | |  | |
| 25. REPORTING REQUIREMENTS (DESCRIPTION:)  Results will be reported in periodic highlights to the U. S. Department of Energy and in journals and conference proceedings. | | | | | | | | | |

c. Purpose

The next generation of computational science applications will require numerical solvers that are capable of high performance on proposed HPC platforms. In order to meet this goal, solvers must be resilient to soft and hard system failures, provide high concurrency on heterogeneous hardware configurations, and retain numerical accuracy and efficiency. In light of these requirements, a natural avenue of inquiry would be to adapt the current stable of numerically efficient solvers to this new high-performance computing regime. However, an alternative approach would be to investigate different classes of algorithms that can address issues of resiliency, particularly fault tolerance and hard processor failures, naturally. In this proposal we will investigate new stochastic methods for solving linear systems, otherwise termed Monte Carlo Resilient, Exascale (MCREX) solvers. The family of methods that we have proposed builds on the sequential Monte Carlo work of Halton, 1962. While showing significant promise, this class of solvers has not made inroads into the broader computational science community. The methods that we have initially developed use Monte Carlo to accelerate a fixed-point iteration; therefore, we have called them Monte Carlo Synthetic Acceleration (MCSA). Preliminary work using MCSA has demonstrated that they are at least as efficient as Jacobi-preconditioned Conjugate Gradient (PCG) on sparse, SPD systems. These initial results demonstrate that very good efficiency could be attained on non-symmetric systems; thus making MCSA an ideal solver in non-linear Newton schemes. Furthermore, Monte Carlo methods have the benefit of addressing resiliency in a natural way; soft errors can be treated as high variance samples and lost histories from processor failures can be easily discarded without affecting the quality of the solution.

d. Background

In nearly all computational engineering and physics fields, linear and non-linear solvers form the core components of modeling and simulation applications. Recent focus on multiphysics coupling adds additional complexity to common linear and non-linear systems as solution strategies change when physical models are coupled. Furthermore, a desire for predictive simulations to enhance the safety and performance of engineered systems creates a need for extremely high fidelity computations to be performed for these coupled systems as a means to capture effects not modeled by coarser methods. In order to achieve this high fidelity, state-of-the-art computing facilities must be leveraged in a way that is both efficient and considerate of hardware-related issues. As scientific computing moves towards exascale facilities, with machines of O(1,000,000) cores already coming online, new algorithms to solve these complex problems must be developed to leverage this new hardware. Issues such as resiliency to node failures and scaling to large numbers of heterogeneous computing elements (CPUs and GPUs) will be pertinent to robust algorithms aimed at this new hardware.

A natural path of investigation in addressing these issues would be to analyze and adapt the existing class of state-of-the-art solvers (Krylov, multigrid, etc.) in order to make them efficient and resilient on current and future hardware. Instead, we pose the question, “Do algorithms exist that naturally enable resiliency while preserving accuracy, convergence, and robustness?” Our objective is to answer this question by proposing a novel group of stochastic methods to advance solution techniques for linear and non-linear problems with a focus on resiliency against both hard and soft errors.

The idea of using Monte Carlo methods (random walks) to invert linear systems is not new. The earliest referenceable work dates to a 1950 paper by Forsythe and Leibler. Their paper credits the idea to unpublished work by J.v. Neumann and S.M. Ulam dating back to the 1940's. The basic principles of Monte Carlo matrix inversion were further elucidated in Hammersley and Handscomb's 1964 text. These early methods are distinguished by very slow, statistically noisy convergence properties; thus, they have not made any significant impact in the linear solver community.

In order to address the convergence issues plaguing Monte Carlo solvers, Halton proposed a Sequential Monte Carlo method. This algorithm demonstrated dramatically improved convergence over regular Monte Carlo. Nonetheless, this method has not gained widespread use as a production-quality linear solver. For many decades, the particle transport community has been utilizing Monte Carlo methods for the solution of transport problems. The partial differential equation (PDE) community has focused on various deterministic methods for solutions to linear problems. In between these two areas are a not widely known small group of stochastic methods for solving sparse linear systems. In recent years, we have further developed these methods for transport problems in the form of Monte Carlo Synthetic-Acceleration (MCSA) that have yet to be applied to more general sparse linear systems. Compared to other methods in these regimes, MCSA offers three attractive qualities: (1) the linear operator need not be symmetric or positive-definite, thereby reducing preconditioning complexity, (2) parallelization using modern methods developed by the transport community is possible, and (3) the stochastic nature of the solution method provides a natural solution to the issue of resiliency.

In addition to linear solver advancements, non-linear solvers may also benefit from a general and parallel MCSA scheme. In the engineering community, non-linear problems are often addressed by linearizing the problem and using traditional iterative or direct methods. In the mathematics community, various Newton methods have been popular. Recently, Jacobian-Free Newton-Krylov (JFNK) schemes have been utilized in multiphysics and advanced single physics codes. The benefits of JFNK schemes are that the Jacobian is never formed, simplifying the implementation, and a Krylov solver is leveraged (typically GMRES or Conjugate Gradient), providing excellent convergence properties for well-conditioned and well-scaled systems. However, there are two potential drawbacks to these methods for high fidelity predictive simulations: (1) the Jacobian is approximated by a first-order differencing method on the order of machine precision such that the error can grow beyond that of those in a fine-grained system and (2) for systems that are not symmetric positive-definite (which will be the case for most multiphysics systems and certainly for most preconditioned systems) the Krylov subspace generated by methods utilizing a long recurrence relation such as GMRES may become prohibitively large. To address these issues, this work proposes novel methods for non-linear systems based on the MCSA method. Although the Jacobian must be explicitly formed to use MCSA, for problems that take more than a few GMRES iterations to converge the storage required for the Krylov subspace will likely grow beyond that of the Jacobian. Finally, using MCSA for the linear solve provides its benefits for preconditioning, parallelism, and resiliency.

e. Approach

We plan to analyze and extend MCSA in order to achieve numerical efficiency, robustness, and accuracy combined with resiliency and performance on current and future HPC architectures. The objective is to obtain resiliency in a manner that naturally fits into the complete MCSA algorithm. Numerical characterization and improvements to MCSA will be performed using the tools of standard numerical analysis. We intend to verify solver resiliency through fault injection. Likewise, we will extrapolate performance estimates to proposed future HPC architectures by developing a performance model.

Numerical investigations of MCSA will be performed in two phases (a) demonstrating convergence, accuracy, efficiency, and robustness, and (b) determining the applicability of subspace iteration schemes. Algorithmic investigations will be focused on improving parallel performance. Both numerical and algorithmic research will investigate resiliency aspects of the solution process. We define resiliency as the ability to recover from both hard and soft errors. Soft errors consist of logical, floating-point, and other processor evaluation faults short of process failure. Hard errors are full process failures. We will show how we can address each failure type algorithmically through numerical analysis and parallel redundancy.

The first stage of our numerics research will be demonstrating convergence, accuracy, efficiency, and robustness of the MCSA algorithm. Preliminary results indicate that the fundamental solver converges and is efficient for sparse SPD systems. However, significant analysis remains to determine

* its convergence properties on non-symmetric systems;
* its effectiveness as the linear solver in non-linear Newton methods;
* an analytical framework that relates weight cutoff and number of histories to problem parameters;
* the effect of high variance events on convergence and robustness.

All of our preliminary studies have focused on sparse SPD systems, in which we have shown MCSA to be competitive with PCG. Because MCSA has no requirements on symmetry or positive-definiteness, it is anticipated that the method will compare favorably to GMRES, BICGSTAB, or TFQMR. In general, deterministic and Monte Carlo solution methods have competing requirements with regard to achieving concurrency; efficient parallelism in deterministic methods is often achieved by decomposing the global phase space as much as possible, whereas in Monte Carlo efficiency is often the result of replicating as much of the phase space as possible. An additional constraint beyond pure performance considerations is the best decomposition strategy that can aid resiliency, particularly to hard system errors. Balancing these requirements, we propose to use a recently developed algorithm for Monte Carlo called Multi-Set, Overlapping Domain (MSOD).

The resilience of the proposed algorithm will be modeled though a series of fault injection campaigns. The impact of a hard or soft error on the algorithm will be investigated by artificially creating such errors during algorithm execution and by observing the resulting runtime behavior, execution time, and output. The two most likely errors in an extreme-scale system are a process failure, e.g., a detected but unrecoverable error leading to process termination, and silent data corruption (SDC), e.g., an undetected bit flip in a memory latch. Both error types will be injected based on their probability, i.e., with a certain frequency and probability distribution, targeting specific algorithm vulnerabilities.

The error probability will be modeled using historical log data from ORNL systems and assumptions communicated by HPC vendors, such as Cray and Intel, regarding the reliability of future-generation systems. The algorithm vulnerabilities will be modeled using a whitebox analysis for identifying the most vulnerable execution paths and data structures in combination with runtime profiling for identifying system component (processor, memory, and network) usage. Correlating algorithm vulnerabilities with system component usage provides a number of injection points at which a specific error class is most likely and is expected to result in a significant impact, such as a process fault during a critical collective operation or SDC in the most significant bits of an important value. The fault injection campaigns will utilize different existing technologies to model the resilience of the algorithm on today's and on future-generation HPC systems.

The fault injection campaigns will provide the data points needed to model the resilience of the proposed algorithm, such that the probability of computing a correct result within a certain time to solution can be estimated for both error classes depending on the reliability properties of the HPC system, the scale of the application run, and the problem size.

For investigation of hard errors on today's HPC systems, a fault-tolerant MPI solution will be used to implement a fully functional resilient algorithm atop a message passing runtime environment that is capable of detecting and surviving process failures and notifying an application about each failure. Injecting a hard error is as simple as manually killing a process. The fault-tolerant MPI automatically reconfigures to survive the process failure and notifies the application about the failure. The notification can be used to trigger any reconfiguration of the application, including algorithm-level recovery. The PIs have access to two different fault-tolerant MPI solutions, FT-MPI developed by the University of Tennessee in 2001 and the prototype for the MPI 3.x standard with fault tolerance support developed by ORNL in 2011. This fault injection campaign will observe the impact of hard errors on the algorithm's execution time and result.

For investigation of soft errors on today's HPC systems, the redundant MPI implementation, redMPI, will be used to perform comparative studies at runtime. redMPI transparently executes an application in a redundant fashion by utilizing the MPI performance tool interface, PMPI, to transparently intercept MPI calls from an application and to hide all redundancy-related mechanisms. A redundantly executed application runs with $r\*m$ native MPI processes, where $r$ is the number of MPI ranks visible to the application and $m$ is the replication degree.

Under normal operation, messages between redundant nodes are replicated and compared for hard and soft error detection and correction. For fault injection campaigns, the message replication and comparison protocol performs detection only. This permits the original parallel application to be tainted with data corruption, while the fully redundant parallel application serves as a correct live control for close observation of the error impact, including its propagation, detection, and masking. Differences between the tainted execution and the control are detected and recorded at runtime by the message replication and comparison protocol, providing detailed information about error sensitivity and propagation.

For investigation of hard and soft errors on future-generation HPC systems, the Extreme-scale Simulator (xSim) performance investigation toolkit will be utilized. xSim is a recently developed performance investigation toolkit that permits running HPC applications in a controlled environment with millions of concurrent execution threads. It allows observing application performance in a simulated extreme-scale system for hardware/software co-design. Using a lightweight parallel discrete event simulation, xSim executes an application on a much smaller HPC system in an oversubscribed fashion with a virtual wall clock time, such that performance data can be extracted based on a processor and a network model with the appropriate simulation scalability/accuracy trade-off. xSim is designed like a traditional performance tool, as an interposition library that sits between the MPI application and the MPI layer, using the MPI performance tool interface. It currently holds the world record in extreme-scale simulation, running up to 134,217,728 communicating MPI tasks, each with its own process context, using just a 960-core Linux-based cluster.

To insure the success of the MCREX solvers on exascale hardware, a vigorous effort will be undertaken to model the algorithm and code performance characteristics as part of the algorithm design process. This work will take place in the setting of the following interrelated steps.

**Algorithm Development**: The MCREX solver algorithms will be designed with concern not only for the algorithm numerics and convergence properties but also with anticipated future hardware characteristics in mind, such as the presence of hard and soft errors, increasing time and power costs of communicating off-die and off-node, decreasing relative sizes of high-speed cache memories and register files, and the need to expose increasing amounts of thread-level parallelism.

**Code Implementation**: The algorithms will be implemented in software, initially as prototypes and then as more full-fledged versions, to be evaluated and tested on current state-of-the-art HPC hardware such as the ORNL Titan system and follow-on hardware.

**Performance Analysis**: The performance characteristics of these codes will be determined, using tools such as VAMPIR, CrayPat and other profiling tools to understand the performance hot spots of the algorithms and understand how the performance depends on characteristics of the different system hardware components.

**Performance Modeling**: Quantitative models will be developed to predict the performance of the algorithms for different problem types of interest under the assumption of multiple potential future architecture scenarios, based on possible futures for exascale hardware. These findings will not only provide a feedback loop for further algorithm design but also provide an assessment of the relative effectiveness of alternative system hardware designs for algorithms of this type. The fully developed performance model will be simulated using the performance investigation features of the xSim toolkit.

The tasks for each year of the proposal are as follows:

|  |  |  |
| --- | --- | --- |
| 1 | Derive and implement linear ADR model equations for MCSA algorithm | Y1 |
| 2 | Investigate solver runtime parameters and performance for ADR model system | Y1, Y2 |
| 3 | Analyze robustness of unbiased estimators to high-variance events | Y1, Y2 |
| 4 | Implement MSOD parallel algorithm | Y2 |
| 5 | Develop performance model for MSOD/MCSA algorithm | Y2, Y3 |
| 6 | Model algorithm resiliency using fault injection campaigns | Y2, Y3 |
| 7 | Estimate algorithm performance on future systems using the performance model and xSim | Y3 |

j. Milestones

**Year 1:** Demonstrate convergence properties and performance of MCSA algorithm on sparse symmetric and non-symmetric systems.

**Year 2:** Show resiliency of MCSA algorithm for soft and hard errors while solving the linear advection-diffusion-reaction (ADR) model problem.

**Year 3:** Show parallel performance of MCSA algorithm on existing HPC architectures and demonstrate scaling to exascale systems using the MCSA performance model.

o. Other

(1) OBLIGATIONS FOR OPERATING EXPENSES-Budget Authority (B/A)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Obligation Estimates | | | | |
|  | FY 2013 |  | FY 2014 |  | FY 2015 |
| Cost (B/O) Estimates | 300 |  | 300 |  | 300 |
| Less: Uncosted Balance (--) at 10/01 | 0 |  | 0 |  | 0 |
| Plus: Commitments for Continued Operations | 0 |  | 0 |  | 0 |
| Outstanding Commitment Balance | 0 |  | 0 |  | 0 |
| TOTAL OBLIGATIONS--CHANGE | 300 |  | 300 |  | 300 |

(2) CAPITAL EQUIPMENT OBLIGATIONS AND COSTS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Obligation and Cost Estimates | | | | | | | | | | | | |
|  | | FY 2013 | | | | |  | FY 2014 | | |  | FY 2015 | | |
| Equipment Items (List) |  | Beginning Uncosted Balance |  | Oblig |  | Cost |  | Oblig |  | Cost |  | Oblig |  | Cost |