Quantitative Modeling of Data Structure Vulnerability With an Application-Centric Approach

Dong Li[†] and Jeffrey S. Vetter[†] [⋆]

[†]Oak Ridge National Laboratory
[⋆]Georgia Institute of Technology
lid1@ornl.gov (main contact), vetter@computer.org

1 Motivation

Resilience remains as one of the major cross cutting design goals for high-end computing systems. Looking forward to Exascale, members of the community expect that both the sheer scale of components, and the move toward heterogeneous architectures, near-threshold computing, and aggressive power management will compound the resiliency challenge.

Today's petascale systems use a combination of hardware, firmware, and system software to solve resilience challenge (e.g., system level checkpoint/restart [DHR02], process replication [FSI+11, FME+12], hardware ECC [Del97, UMB+12] and virtualization [NMES07, VNE+08]). Those existing resilience mechanisms usually apply a monolithic approach and tend to be rigid. As a result, system resilience comes with large performance and power costs. One of the core reasons for this situation comes from the lack of sufficient understanding of application vulnerabilities. In particular, we cannot easily quantify application resilience; hence, it is very difficult to decide which resilience mechanism is the most suitable and how to make resilience mechanisms and architecture adaptive to what the application needs. Although using a resilience-aware programming model [CLS+12] to explicitly structure applications for the convenience of fault detection and location can partially solve this problem, the programming model requires application specific knowledge from the programmer to indicate application vulnerability, and it requires significant effort to re-factor applications.

In this position paper, we propose to quantify vulnerability of data structures within the application. This quantitative modeling will greatly advance our understanding of application characteristics from the perspective of resilience. As a result of our method, future research should be able to implement adaptive, fine-grained data protection, potentially improving performance and power efficiency by reducing protection costs. Furthermore, the proposed quantitative modeling can be beneficial for cross-layer coordination and optimization, which avoids data over-protection and further improving system efficiency.

2 Our Position

We choose to quantitatively model vulnerability of the data structure, and develop new *machine and application abstractions* based on this model. Since the data structure is the fundamental protection target of many resilience mechanisms (e.g., application-level checkpointing/restart, hardware ECC and algorithm-based fault tolerance), our modeling result will be easy to integrate with these mechanisms, enabling *dynamic and actionable modeling*.

The vulnerability of a data structure is determined by the amount of fault masking in a system. The fault masking can be provided by the application or the architecture. From the application aspect, depending on memory access patterns and application algorithms, a fault in a specific data structure may not cause user-visible results. For example, if an error in a data structure is self-contained and never propagates to application-critical states, or if the error can be easily averaged out in specific execution phases (e.g., iterative solvers), then it may not impact the application results. From the architecture perspective, at any given cycle, a fault in the target data structure may not propagate to the inputs of an architecture component within its window vulnerability, and, hence, may not cause user-visible errors.

To account for the above fault masking effects, we develop a binary instrumentation based fault injection tool to classify data structures based on their vulnerability [LVY12]. Furthermore, we rely on the architecture simulation, and use application knowledge (e.g., application semantics) to direct simulation. This approach allows us to identify hardware events related with the target data structure; more importantly, we are able to include the effects of application-level protection (e.g., checkpointing and algorithm-level fault tolerance) during the simulation. Therefore, this work can enable the possibility of *integration and interoperation* of multiple resilience methodologies and tools.

3 Related Work

Previous research employs architecture-centric approaches to quantify architecture vulnerability [BRE+05, NJE10, WHG07, DLP09, SK09, TGLF11]. Essentially, these approaches compute the probability that a fault in a particular (micro)architecture will result in a user-visible error. These research efforts have no ability to model vulnerability of data structures, and cannot provide practical, actionable utility in resilience design for applications and architectures.

The traditional approach to analyze application vulnerability is fault injection [BdS08, SSR11, MRK10, DBG+11, NBV+09, LR04, LVY12]. This approach usually employs bit-flipping at architectural, gate-level, or high level application states. By injecting a large number of representative corruptions into the application, the fault injection statistically reveals the application vulnerability. However, this method provides only limited fault coverage because of extremely large cost of exhaustive fault injection. Also, the statistical nature of this method provides unbounded inaccuracy, and may lead to an incorrect resilience design.

4 Assessment

Challenges addressed: The proposed approach addresses resilience and power challenges for future exascale systems as identified by DOE. The ability to improve application resilience with lower performance and power overhead will be critical to improve the scalability of the exascale systems. The proposed approach identifies key areas of co-design where ModSim can have a significant impact; the proposed approach also identifies research and development areas that require joint efforts from multiple layers of the system stack.

Maturity: The proposed idea builds on successful research, including the DOE funded Blackcomb for Advanced Architectures and Critical Technologies for Exascale Computing, and CESAR co-design efforts. We have preliminary capabilities to classify data structure vulnerability based on random fault injection [LVY12]. We have developed simulation capabilities to perform resilience and architecture research [LVM+12, LCWV13].

Uniqueness: The resilience challenge addressed in this paper is unique to the exascale systems. Although the general methodology proposed in this paper could be applied to other compute environment, the unprecedented scale and power concerns in exascale systems is not likely appeared in other areas or communities.

Novelty: The resilience community of HPC does not have any quantitative and accurate approach to understand application characteristics in terms of resilience (especially data structure vulnerability). Essentially all current approaches attempt to rely on statistical-based fault injection, which have limited capabilities for fault coverage and accuracy. Our approach attempts to solve these problems.

Applicability: The proposed approach can be applied to profile DOE applications and provides valuable guidance for application development and architectures. It can also be used to studying interaction effects between performance, resilience and power.

Effort: Investigating this approach is likely to be a multi-year effort. It will require several FTEs to target architecture features and application characteristics for modeling, implement modeling capabilities based on existing simulation platforms, evaluate modeling, and improve simulation with realistic DOE applications.

References

[BdS08] G. Bronevetsky and B. de Supinski. Soft Error Vulnerability of Iterative Linear Algebra Methods. In *International Conference on Supercomputing*, 2008.

[BRE+05] Arijit Biswas, Paul Racunas, Joel Emer, Shubhendu S. Mukherjee, and Ram Rangan. Computing Architectural Vulnerability Factors for Address-Based Structures. In *International Symposium on Computer Architecture*, 2005.

[CLS+12] Jinsuk Chung, Ikhwan Lee, Michael Sullivan, Jee Ho Ryoo, Dong Wan Kim, Doe Hyun Yoon, Larry Kaplan, and Mattan Erez. Containment Domains: A Scalable Efficient and Flexible Resilience Scheme for Exascale Systems. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2012.

[DBG+11] Nathan DeBardeleben, Sean Blanchard, Qiang Guan, Ziming Zhang, and Song Fu. Experimental Framework for Injecting Logic Errors in a Virtual Machine to Profile Applications Soft Error Resilience. In *Workshop on Resilience in High Performance Computing in Clusters, Clouds and Grids*, 2011.

[Del97] T. Dell. A White Paper On The Benefits Of Chipkill-Correct ECC for PC Server Main Memory. Technical report, IBM Microelectronics Division, 1997.

[DHR02] J. Duell, P. Hargrove, and E. Roman. The Design and Implementation of Berkeley Lab's Linux Checkpoint/Restart. Technical report, Berkeley Lab, 2002.

[DLP09] Lide Duan, Bin Li, and Lu Peng. Versatile Prediction and Fast Estimation of Architectural Vulnerability Factor from Processor Performance Metrics. In *International Symposium on High Performance Computer Architecture*, 2009.

[FME+12] David Fiala, Frank Mueller, Christian Engelmann, Rolf Riesen, Kurt Ferreira, and Ron Brightwell. Detection and Correction of Silent Data Corruption for Large-Scale HPC. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2012.

[FSI+11] Kurt Ferreira, Jon Stearley, James H. Laros III, Ron Oldfield, Kevin Pedretti, Ron Brightwell, Rolf Riesen, Patrick G. Bridges, and Dorian Arnold. Evaluating the Viability of Process Replication Reliability for Exascale Systems. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2011.

[LCWV13] Dong Li, Zizhong Chen, Panruo Wu, and Jeffrey S. Vetter. Rethinking Algorithm-Based Fault Tolerance Using a Software-Hardware Cooperative Approach. In *Under submission*, 2013.

[LR04] C. Lu and D. Reed. Assessing Fault Sensitivity in MPI Applications. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2004.

[LVM+12] Dong Li, Jeffrey S. Vetter, Gabriel Marin, Collin McCurdy, Cristi Cira, Zhuo Liu, and Weikuan Yu. Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications. In *International Symposium on Parallel and Distributed Processing*, 2012.

[LVY12] Dong Li, Jeffrey S. Vetter, and Weikuan Yu. Classifying Soft Error Vulnerabilities in Extreme-Scale Scientific Applications Using a Binary Instrumentation Tool. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2012.

[MRK10] K. Malkowski, P. Raghavan, and M. Kandemir. Analyzing the Soft Error Resilience of Linear Solvers on Multicore Multiprocessors. In *International Symposium on Parallel and Distributed Processing*, 2010.

[NBV+09] Thomas Naughton, Wesley Bland, Geoffroy Vallee, Christian Engelmann, and Stephen L. Scott. Fault Injection Framework for System Resilience Evaluation: Fake Faults for Finding Future Failure. In *Workshop on Resilience in High Performance Computing*, 2009.

[NJE10] Arun Nair, Lizy John, and Lieven Eeckhout. AVF Stressmark: Towards an Automated Methodology for Bounding the Worst-case Vulnerability to Soft Errors. In *International Symposium on Microarchitecture*, 2010.

[NMES07] Arun B. Nagarajan, Frank Mueller, Christian Engelmann, and Stephen L. Scott. Proactive Fault Tolerance for HPC with Xen Virtualization. In *International Conference on Supercomputing*, 2007.

[SK09] Vilas Sridharan and David R. Kaeli. Eliminating Microarchitectural Dependency from Architectural Vulnerability. In *International Symposium on High Performance Computer Architecture*, 2009.

[SSR11] M. Shantharam, S. Srinivasmurthy, and P. Raghavan. Characterizing the Impact of Soft Errors on Iterative Methods in Scientific Computing. In *International Conference on Supercomputing*, 2011.

[TGLF11] Jingweijia Tan, Nilanjan Goswami, Tao Li, and Xin Fu. Analyzing Soft-Error Vulnerability on GPGPU Microarchitecture. In *International Symposium on Workload Characterization*, 2011.

[UMB+12] Aniruddha N. Udipi, Naveen Muralimanohar, Rajeev Balsubramonian, Al Davis, and Norman P. Jouppi. LOT-ECC: Localized and Tiered Reliability Mechanisms for Commodity Memory Systems. In *International Symposium on Computer Architecture (ISCA)*, 2012.

[VNE+08] Geoffroy Vallee, Thomas Naughton, Christian Engelmann, Hong Ong, and Stephen Scott. System-level Virtualization for High Performance Computing. In *Euromicro International Conference on Parallel, Distributed, and network-based Processing*, 2008.

[WHG07] Kristen Walcott, Greg Humphreys, and Sudhanva Gurumurthi. Dynamic Prediction of Architectural Vulnerability from Microarchitectural State. In *International Symposium on Computer Architecture*, 2007.