Resiliency in Exascale Systems and Computations Using Chaotic-Identity Maps

Nageswara S.V. Rao

Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
raons@ornl.gov

Abstract. For exascale computing systems, we propose (i) light-weight computational modules that utilize chaotic computations and customized identity maps to detect component failures, and (ii) statistical estimation methods that generate robustness estimates for the system and computations based on the module outputs. The diagnosis modules execute multiple Poincare and identity maps, which are customized to detect certain classes of failures in the compute nodes and interconnects. We propose statistical methods that generate robustness estimates for the system using the outputs of pipelined chains of diagnosis modules.

Keywords: Exascale systems, resiliency, chaotic maps, statistical estimation.

1 Introduction

Exascale computing systems are expected to have processor cores, memory units, communications and other components totaling in the numbers of millions [2]. Based on a life-span of about ten years for the components, computations of a few hours are likely to experience several component failures. As a result, outputs of codes on these systems cannot be guaranteed to be failure-free, even if failures are detected and corrected using methods such as checkpoint recovery. But such failures could have a significant impact on scientific computations. The errors could propagate through computations, such as hydrodynamic or climate codes, and potentially lead to incorrect outputs. A strategy to mitigate such effects must combine efforts to make the system robust (by methods such as checkpointing and fault-tolerant hardware), and also to enhance application codes with software fault tolerance methods.

System and application software can be designed to be robust against faults in the computing system, by utilizing combinations of methods such as checkpointing, N-version programming, process migration and more specific methods such as fault oblivious algorithms and fault tolerant MPI. However, their effectiveness drops-off exponentially with the failure levels of the system. On the other hand, it is very unproductive to design them for the worst-case failure mode, since it could degrade the computational productivity to levels of much smaller systems. Therefore, it is critical to estimate the current robustness level

of the computing system to guide these software solutions, and also to assign confidence levels to the code outputs.

From a broad perspective, methods for achieving robust computations on failure-prone computing systems have been developed in several cases. von Neumann [6] studied the mathematical aspects of building reliable computing systems from unreliable components in the 50's. These works establish analytically that it is possible to achieve robust computations even if the computing platform is inherently failure-prone. In terms of practical systems, space vehicles deployed over the past decades have been enhanced with Software-Implemented Hardware Fault Tolerance (SIHFT) methods to counteract the transient computing faults that arise due to radiation exposure. These works show that it is indeed possible to build robust computing systems capable of operating in some of the most challenging, failure-prone environments.

1.1 Our Approach

We address a class of problems of assessing the robustness levels of an exascale system at the system- and application-levels under certain failures in the compute nodes and interconnects. Our approach is based on a synthesis of methods from fault diagnosis [3] and chaotic Poincare maps [1], and consists of: (a) diagnosis methods to identify the errors due to failures using strategically guiding the execution paths, (b) Poincare maps to amplify their effects making them quickly detectable, and (c) statistical estimation methods [5] to process the data from executional traces to generate system robustness profiles and confidence estimates for application computations. The same basic methods are applied for both tasks: (i) in a strategic, pipelined configuration to estimate the system robustness, and (ii) embedded in application codes to sample their executional paths and generate confidence measures. The chaotic maps are chosen for their low computational requirements and sensitivity to computational errors arising from a wide variety of faults, including those in Arithmetic and Logic Unit (ALU), memory units and interconnects. In this extended abstract, we only briefly described our approach, and more details including preliminary proof-of-principle illustrations and simulations are provided in [4].

References

1. Alligood, K.T., Sauer, T.D., Yorke, J.A.: *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag Pub., Reading (1996)
2. International exascale software project, <http://www.exascale.org>
3. Rao, N.S.V.: Computational complexity issues in operative diagnosis of graph-based systems. *IEEE Transactions on Computers* 42(4), 447–457 (1993)
4. Rao, N.S.V.: Chaotic-Identity Maps for Robustness Estimation of Exascale Computations. In: *IEEE Workshop on Fault Tolerance for HPC at Extreme Scale* (2012)
5. Vapnik, V.N.: *Statistical Learning Theory*. John-Wiley and Sons, New York (1998)
6. von Neumann, J.: Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon, C.E., McCarthy, J. (eds.) *Automata Studies*. Princeton University Press (1956)