The International Conference for High Performance Computing,
Networking, Storage and Analysis (SC14)
Doctoral Showcase Program

Research Summary

Saurabh Hukerikar
saurabh@isi.edu

Dissertation Topic: Introspective Resilience for Exascale High Performance Computing Systems
Dissertation Adviser: Dr. Robert F. Lucas
Expected Completion: May 2015

Summary:
By the end of the decade, exascale-class High Perfomance Computing (HPC) systems promise to push the frontiers of scientific and engineering research by enabling the solution of vastly more accurate predictive models and the analysis of massive data sets. These next generation systems, capable of performing a quintillion (10^{18}) operations per second, will seek to address complex problems through simulation and modeling of physical phenomena. These systems will be architected through the deployment of millions of ALUs and memory chips. With the scaling of VLSI geometries, semiconductor devices will be increasingly susceptible to errors and probabilistic in their behavior. Even with optimistic assumptions on the reliability of chips, the massive scale of the systems amplifies the problem of system unreliablity. In these systems, faults will become the norm, not the exception and long-running scientific applications will increasingly experience disruptions.

Today's model of execution for HPC systems completely abstracts the underlying fabric of hardware and system software, such that the application layer can always assume correct behavior. The most widely used techniques for managing application resilience are based on Checkpoint and Rollback (C/R) which activate only upon failure of a process.

Recovery is performed by restarting the application from the latest global checkpoint. This approach may prove ineffective when the system mean time to failure (MTTF) is lower than the time interval required to create and commit a checkpoint or the time needed to restart from a stable checkpoint in persistent storage. While algorithmic techniques offer the ability to detect and correct bit flip errors in part of the application state for certain numerical problems, they require encoding the data structures and adapting the algorithms to operate on the encoded data. However, in each of these approaches, there is lack of communication between the layers of system abstraction, notably between the hardware layer, where most of the faults and errors originate, and the application layer and libraries, where the application correctness is affected. There being little coordination between layers of system abstraction amounts to the upper layers of the software stack being insufficiently equipped to cope with the errors.

In this dissertation work, we propose an introspective approach to managing the resiliency of future exascale HPC systems and their applications. Our proposed approach is based on leveraging programmer insight into the fault tolerance features of HPC applications. Through a set of modest extensions to current programming models we capture the programmer's knowledge on the application's fault resilience. This provides insights into what aspects of the application's active state will be fault tolerant. During application execution, this enables cross-layer efforts for error detection, masking and recovery with the support of a runtime system. For applications, this translates to more errors survived and therefore longer Mean Times To Failure (MTTF).

Next, we hypothesize that through automatic compiler transformations, the application code can be equipped with mechanisms to detect/correct certain errors in their active program state. We propose an adaptive Redundant Multithreading (RMT) approach to enable application level fault detection and correction. The redundant execution of certain application code sections is dynamically enabled/disabled by a runtime system based on continuously observing and assessing the fault tolerance state of the system through hardware based indicators. This enables matching the application's requirements to the fault tolerance state of the system. Such a reasoned application of redundant computation enables opportunistic error detection and mitigation at low overhead to the application's time to solution.

We also propose an introspection runtime framework which allows applications to actively search for errors in the program state as well as enables the system stack to become self-aware of its continuously evolving fault tolerance capabilities. This is based on a trend analysis of the fault events that provides an assessment of the system vulnerability. By reasoning about the rates and sources of faults, and their significance to the outcomes of HPC applications, an introspective approach enables flexible, adaptive utilization of resilience strategies. We experimentally evaluate our proposed mechanisms for a range of HPC workloads with accelerated fault injection rates. Our preliminary results demonstrate much promise to meet the reliability demands and expectations of HPC applications on future exascale platforms.