Improving Fault Tolerance for Extreme Scale Systems

Eduardo Berrocal, Zhiling Lan
{eberroca,lan}@iit.edu

Illinois Institute of Technology, Chicago, IL

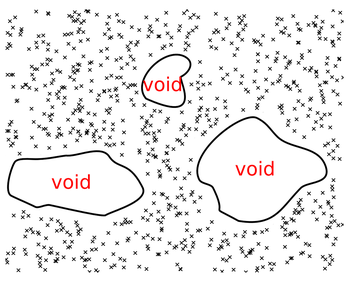
Abstract

The advancement of resilience technologies greatly depends on a deeper understanding of faults arising from hardware and software components. This understanding has the potential to help us build better fault tolerance technologies. In our work we presented a new approach for fault detection based on the Void Search (VS) algorithm, and evaluated the algorithm using real environmental logs from Mira Blue Gene/Q supercomputer at Argonne National Laboratory. Other problems will also arise as transistor size and energy consumption of future systems must be significantly reduced, steps that might dramatically impact the soft error rate (SER). In our previous work we laveraged the fact that datasets produced by HPC applications (i.e., the applications' state at a particular point in time) have characteristics that reflect the properties of the underlying physical phenomena that those applications attempt to model. These characteristics can be used effectively to design a general corruption detection scheme with relatively low overhead. Both of these problems are critical for the HPC community and a main focus of my Ph.D. research.

1.- Hard Error Prediction

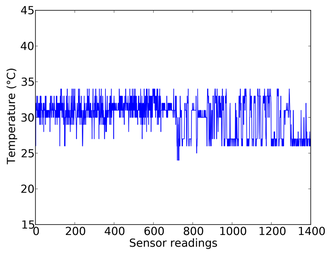
Void Search

- Void Search (VS) are a family of methods aimed at finding regions of empty - or low density - space for a given set of data points.
- These empty regions are known as voids, and in astrophysics they are essential for studying galaxy formation and the structure of the cosmic web.
- For us, voids are essentially patterns of feature space for low density data regions.



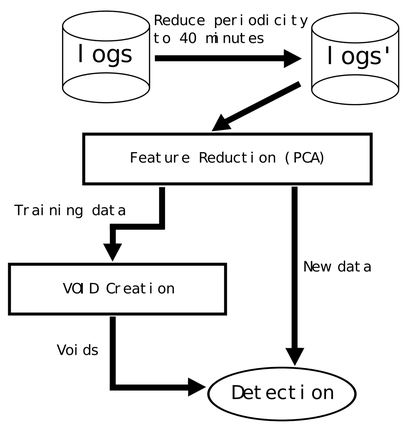
Environmental Data

- Environmental logs are numerical values directly read from the hardware sensors spread all over the system such as fan speed, CPU temperature, input voltage, and so on.
- Since every new generation of supercomputing systems come with better hardware sensors and profiling capabilities, it is of great importance to understand environmental data especially with respect to resilience.



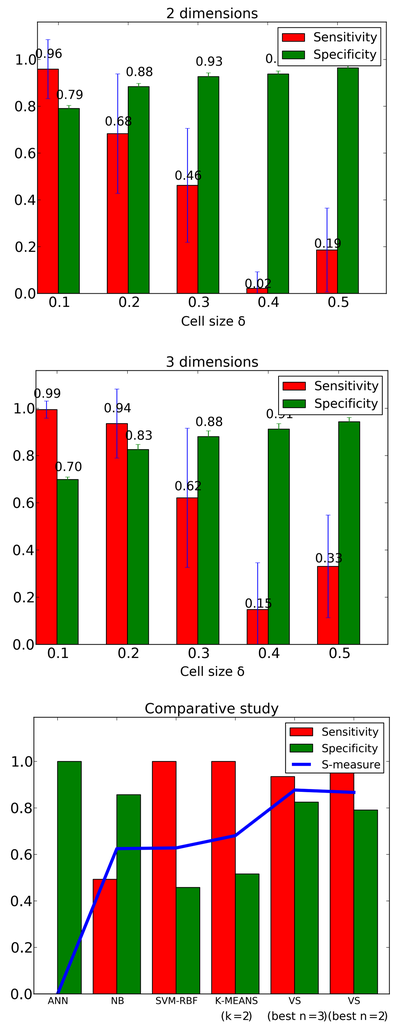
Our Approach

- Our approach is composed of three main steps: step 1 is data pre-processing, particulary feature reduction; step 2 is void construction to build voids for each node; step 3 is to detect faults

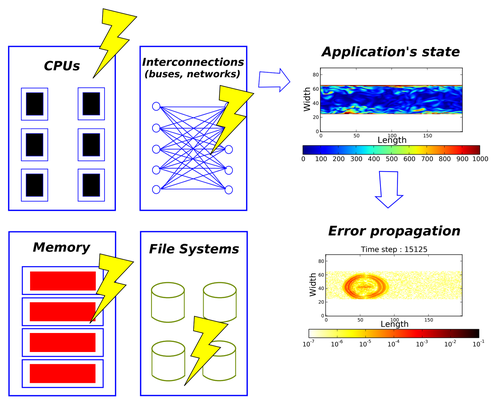


- Our algorithm is node card centric, caring only about faults that can affect node cards directly or indirectly. There are 180 sensors relevant to each

Results



2.- Soft Error Detection



Particle-induced Soft Errors (2012) *

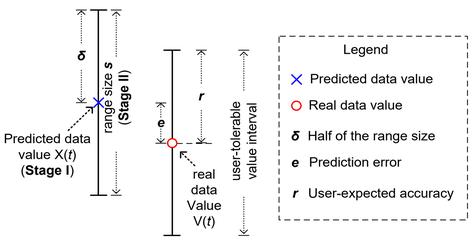
- SRAM: 10^{-4} FIT/bit (1 upset every 10 hours for 1TB)
- Latches: 0.1 - 0.5 FTI/bit
- Logic: 0.1 - 0.5 FTI/bit
- DRAM: 10-20 FIT/device

* M. Snir et. al., "Addressing Failures in Exascale Computing", Report on a Workshop organized by the Institute for Computing Sciences, 2012, Park City, Utah.

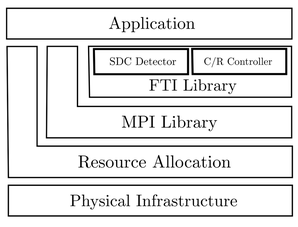
Anomaly Detection

Our pointwise SDC detection model has two phases:

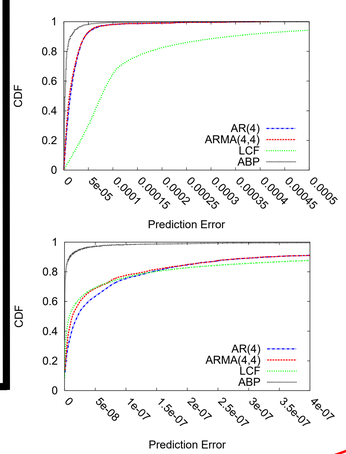
(1) Prediction of the next expected value in the time series for each data point.
(2) determining a range (i.e., normal value interval) surrounding the predicted next-step value.



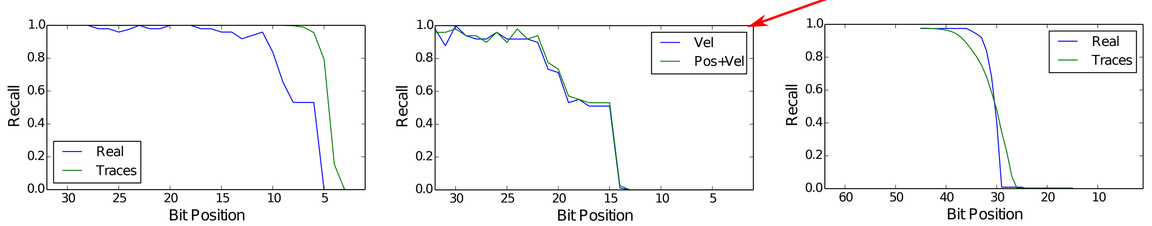
System Overview



Results

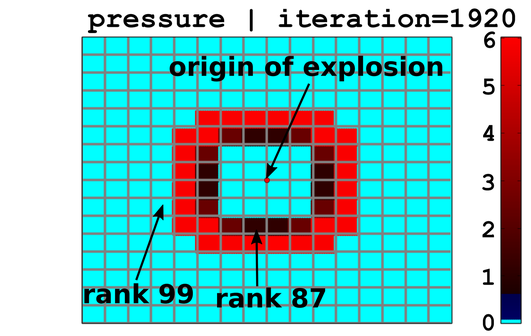


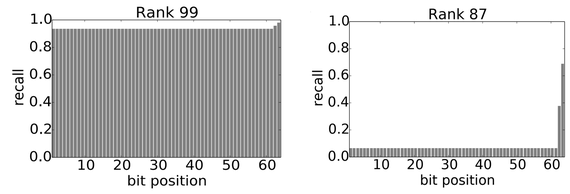
- *A relatively simple predictor such as ABP achieves the smallest prediction errors.*
- It is possible to detect corruptions in one variable (velocity) by using another (position) in HACC by taking advantage of the fact that these variables are interconnected by the underlying physics of the application.



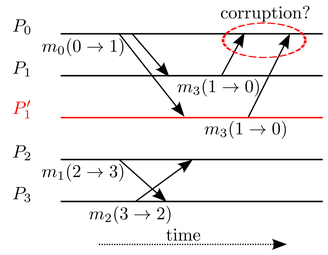
Adaptive Anomaly Detection

- Some applications do not behave smoothly (e.g., explosions, collisions).
- Nevertheless, sharp changes in the data are sometimes concentrated in particular places in space (and/or time).

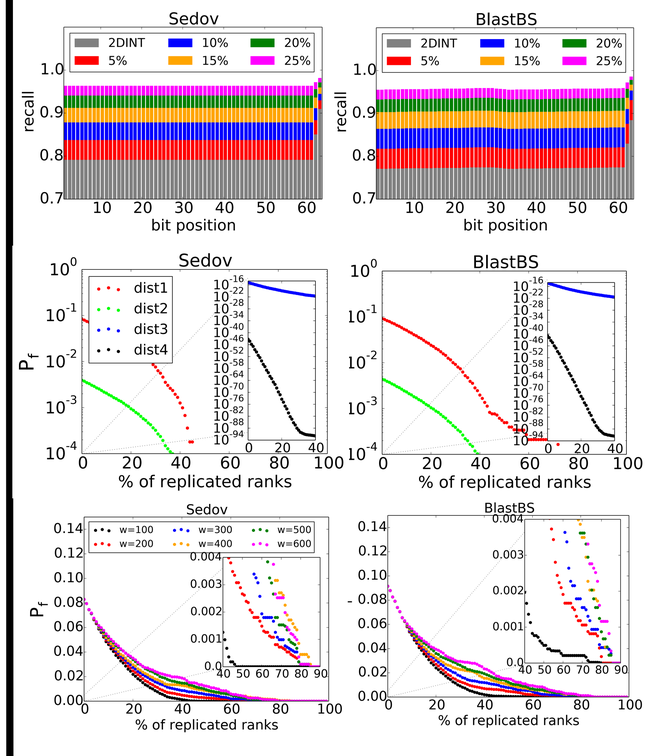




- Our approach combines the merits of replication and data-analytic-based (DAB) detection.
- Partial replication is used for those data regions where our DAB detectors perfom poorly.



Results



Acknowledgments

This material was based upon work supported by the U.S. Department of Energy, Oce of Science, Advanced Scientic Computing Research Program, under Contract DE-AC02-06CH11357, and by the ANR RESCUE and the INRIA-Illinois-ANL- BSC-JSC-Riken Joint Laboratory on Extreme Scale Computing.The work at the Illinois Institute of Technology is supported in part by U.S.National Science Foundation grants CNS-1320125 and CCF-1422009.