AID

AID: Adaptive Impact-Driven Detection library for corruption detection

AID provides a way for HPC users of dynamic simulations over multiple time steps to detect corruptions that impact the results of their execution.

AID is designed to monitor the state data of the application: variables that are the outcome of the execution.

AID is a library offering functions to help programmers defining which variable should be monitored.

AID offers only detection. For recovery we suggest to combine AID with FTI. But AID could be used in combination with any other recovery library.

AID is simple to use:

There are only four steps for users to annotate their MPI application codes:

(1) initialize the detector by calling SDC_Init();

(2) specify the key variables to protect by calling SDC_Protect(var,ierr);

(3) annotate the execution iterations by inserting SDC_Snapshot() into the key loop;

(4) release the memory by calling SDC_Finalize() in the end.

AID supports both C and Fortran.

-->>> **Code download** <<<--

(Contact: sdi1@anl.gov)

If you download the code, please let us know who you are. We are very keen of helping you using the AID library.

A paper describing AID and its detection performance is to appear in Transactions on Parallel and Distributed Systems (TPDS). Its technical report version is available to download.

Spatial Support-vector-machines Detector (SSD)

SSD is a low-memory-overhead effective SDC detector, by leveraging epsilon-insensitive support vector machine regression.

SSD is simple to use, similar to AID, with only four steps for users to annotate their MPI application codes. It supports both C and Fortran interfaces, which are exactly the same as those of AID.

The installation requires Java development kit (JDK), so please make sure JDK is installed well before installing SSD.

-->>> Code download <<<-- (soon, pending DoE approval of distribution licence)

(The code is ready to use, but it cannot be released now because the BSD license is under approval process. Before the official release, the code is available upon request. Contact: omer.subasi@bsc.es)

If you download the code, please let us know who you are. We are very keen of helping you using the SSD library.

A paper describing SSD and it is to appear in CCGrid16.

MAchine-learning based CORruption Detector (MACORD)

MACORD is a new SDC detector for detecting SDCs in HPC applications.

Abstract:

Future HPC systems with ever-increasing resource capacity (such as compute cores, memory and storage) may significantly increase the risks on reliability. Silent data corruptions (SDCs) or silent errors are one of the major sources that corrupt HPC execution results. Unlike fail-stop errors, SDCs are rather harmful and dangerous in that they cannot be detected by hardware. We propose an online MAchine-learning based CORruption Detection framework (abbreviated as MACORD) for detecting SDCs in HPC applications. In particular, we comprehensively investigate the prediction ability of a multitude of machine-learning algorithms in our study, and enable the detector to automatically select the bestfit algorithms at runtime to adapt to the data dynamics. Our learning framework exhibits low memory overhead (less than 1%), since it takes only spatial features (i.e., neighboring data values for each data point in the current time step) into the training data. Experiments based on real-world scientific applications/benchmarks show that our framework can get the detection sensitivity (i.e., recall) up to 99% while the false positive rate is limited down to 0.1% in most cases, which is one order of magnitude improvement compared with the latest state-of-art spatial technique.

(Contact: omer.subasi@bsc.es)