Universidade: presente!

UFRGS PROPESQ

XXXI SIC

Salão UFRGS 2019 – Conhecimento, Formação, Inovação

21 . 25 . OUTUBRO . CAMPUS DO VALE

Non-intrusive Fault Injection Techniques for Efficient Soft Error Vulnerability Analysis

Vitor Bandeira, Ricardo Reis

{vvbandeira,reis}@inf.ufrgs.br

Motivation

- ✔ Multicore processors are de-facto components in many industrial segments, including automotive (e.g., 3 quad-core Arm Cortex-A72)
- ✔ Modern embedded systems are expected to experience at least one soft error per day in the near future, which may lead to life-threatening failures.
- ✔ State-of-the-art frameworks only support the injection of bitflips in memory and general single-core processor components and lack of detailed and customizable post-simulation analysis

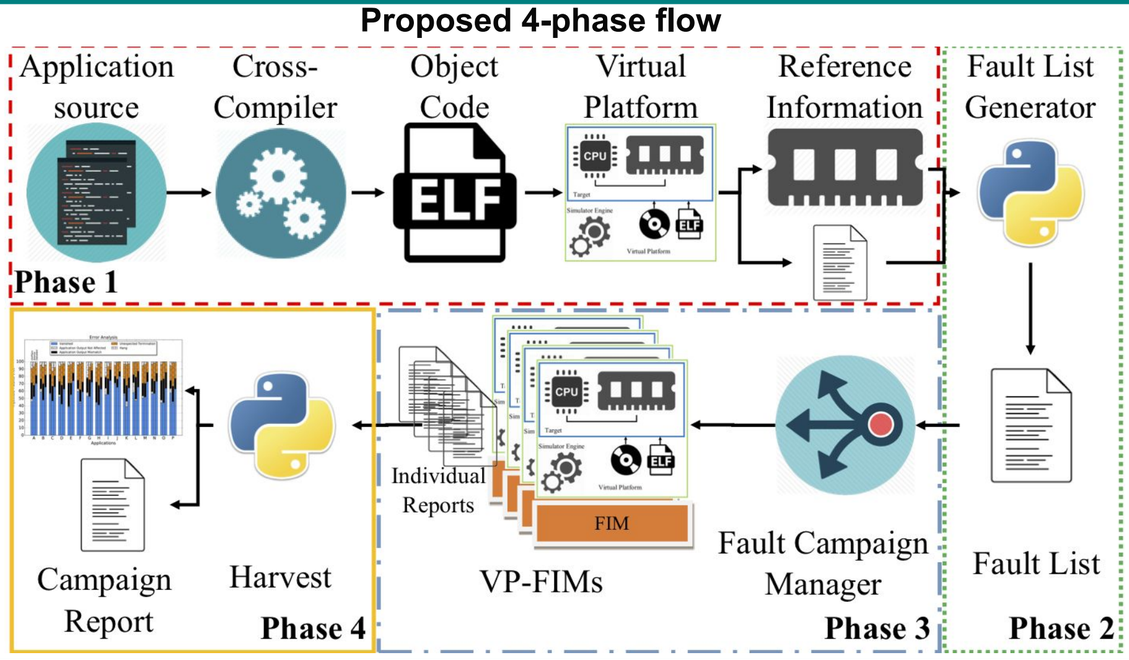
Contributions

1. Detailed observation and analysis of complex automotive vehicle software stacks, including the two real applications found in today's car: object detection and visual odometry
2. High simulation performance enabling large fault injection campaigns varying different aspects that impact on the soft error reliability
3. High fault injection controllability that facilitates the isolation of critical application functions, allowing a detailed analysis of specific critical application, operating system or API structure/functionality

1. Simulation Infrastructure

- ✔ Fault Injector Module (FIM)
  - random bit-flips
  - multicore model support
- ✔ Complex workloads
  - multiple Linux kernels
  - parallelization APIs
- ✔ Open Virtual Platforms (OVP)
  - instruction-accurate
- ✔ Simulation Performance
  - checkpoint
  - host multicore
  - distributed simulation (HPC)

Proposed 4-phase flow



1. model simulation without faults (memory and CPU context collection)
2. fault list creation
3. platform simulation with fault injection. Each application behavior is compared to the golden run and an error report is generated
4. final Report

2. Fault Classification

- ✔ Embedded Classification;
  - Vanished, no fault traces are left;
  - Application Output Not Affected (ONA), the resulting memory is not modified. However, one or more remaining bits of the architectural state is incorrect;
  - Application output mismatch (OMM), the application terminates without any error indication. However, the resulting memory is affected;
  - Unexpected termination (UT), the application terminates abnormally with an error indication;
  - Hang, the application does not finish requiring a preemptive remove.
- ✔ Configurable soft error assessment module
  - Compare internal variables in real time
  - Trace the application control flow

3. Experimental Setup

- ✔ LIBVISO2
  - Library for Visual Odometry
  - Used to determine the vehicle position and orientation by analysing a series of images
  - Two complete sets 11 (920 frames) and 14 (630 frames)
- ✔ Darknet and YOLO
  - Darknet is an open source Neural Network (NN) framework, combined with the YOLO, an image classifier based on convolutional NN which perform real-time object detection