

LA-UR-15-23146

This document is approved for public release; further dissemination unlimited

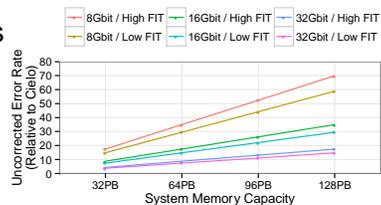
Resilience / Fault Injection Research

Studying Application Response to Faults Paves the Path for Next Generation HPC

BACKGROUND & MOTIVATION

Supercomputing technology is undergoing unprecedented changes that impact resilience of systems and applications.

- Fault characterization provides insight into faults experienced today and allows estimates of the future
- By injecting faults into apps and systems we can learn what to expect and how to improve systems of tomorrow.



INNOVATION

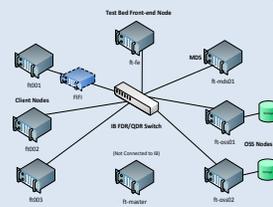
End-to-End Resilience Story and Experimentation Framework

- Take real applications of importance
- Study them under real workloads (full operating systems, middleware, etc.)
- Inject faults with precision at the virtual architecture level.
- Inject faults with precision in the network data.
- How does the application behave in these conditions?
- Campaign studies of fault-tolerance
- Improve the application, empirically study the improvements to make more resilient systems.

DESCRIPTION

Field Data From Today Influences Predictions For Tomorrow and Facilitates Fault Injection Experiments to Understand Application Responses

- Studied DRAM and SRAM faults across 3 DOE supercomputing centers
 - ~50 billion DRAM hours, largest studies of their kind.
 - Directly impacting next generation HPC products
- F-SEFI used to “dial in” real error rates of today and expectations of tomorrow: how do applications of interest fare?
- FIFI is the only network-level I/O fault injector for IB 4x QDR (40 Gb/s)
- Can be placed in system without altering network configuration and applications
- Extremely low latency (< 300 nanoseconds) to minimize system perturbations
- Choice of packets/words/bits for FI made at random or deterministically



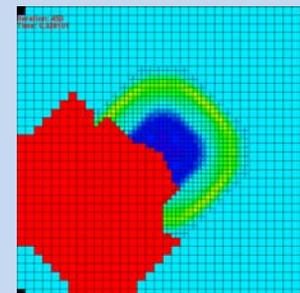
Captured Packet	Correct Output Data	Captured FI Packet	Output Data after FI
Input Data: 000003f0000003f	visibility_cmpqb[0][1]vld[0] = 0.500000000	visibility_cmpqb[0][0]vld[0] = 0.500000000	visibility_cmpqb[0][0]vld[0] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[0] = 0.000000000	visibility_cmpqb[0][0]vld[0] = 0.000000000	visibility_cmpqb[0][0]vld[0] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[1] = 0.500000000	visibility_cmpqb[0][0]vld[1] = 0.500000000	visibility_cmpqb[0][0]vld[1] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[2] = 0.500000000	visibility_cmpqb[0][0]vld[2] = 0.500000000	visibility_cmpqb[0][0]vld[2] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[3] = 0.000000000	visibility_cmpqb[0][0]vld[3] = 0.000000000	visibility_cmpqb[0][0]vld[3] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[4] = 0.500000000	visibility_cmpqb[0][0]vld[4] = 0.500000000	visibility_cmpqb[0][0]vld[4] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[5] = 0.000000000	visibility_cmpqb[0][0]vld[5] = 0.000000000	visibility_cmpqb[0][0]vld[5] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[6] = 0.500000000	visibility_cmpqb[0][0]vld[6] = 0.500000000	visibility_cmpqb[0][0]vld[6] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[7] = 0.000000000	visibility_cmpqb[0][0]vld[7] = 0.000000000	visibility_cmpqb[0][0]vld[7] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[8] = 0.500000000	visibility_cmpqb[0][0]vld[8] = 0.500000000	visibility_cmpqb[0][0]vld[8] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[9] = 0.000000000	visibility_cmpqb[0][0]vld[9] = 0.000000000	visibility_cmpqb[0][0]vld[9] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[10] = 0.500000000	visibility_cmpqb[0][0]vld[10] = 0.500000000	visibility_cmpqb[0][0]vld[10] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[11] = 0.000000000	visibility_cmpqb[0][0]vld[11] = 0.000000000	visibility_cmpqb[0][0]vld[11] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[12] = 0.500000000	visibility_cmpqb[0][0]vld[12] = 0.500000000	visibility_cmpqb[0][0]vld[12] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[13] = 0.000000000	visibility_cmpqb[0][0]vld[13] = 0.000000000	visibility_cmpqb[0][0]vld[13] = 0.000000000
000003f0000003f	visibility_cmpqb[0][0]vld[14] = 0.500000000	visibility_cmpqb[0][0]vld[14] = 0.500000000	visibility_cmpqb[0][0]vld[14] = 0.500000000
000003f0000003f	visibility_cmpqb[0][0]vld[15] = 0.000000000	visibility_cmpqb[0][0]vld[15] = 0.000000000	visibility_cmpqb[0][0]vld[15] = 0.000000000

TRL 5-6: FIFI tested and verified via pilot study/ongoing I/O testing. F-SEFI demonstrated on a host of apps including LANL hydrodynamics mini-app, CLAMR

ANTICIPATED IMPACT

Fault Injection Experiments in Real Applications Provides Insight

- Insights from F-SEFI experiments provide guidance to hydrodynamics app developers.
- Lead to detection and correction work to become resilient to soft errors
- Empirically tested and measured



PATH FORWARD

Implement outside LANL:

- Utilize F-SEFI and FIFI in fault testing to enable end-to-end resilience research
- Future FIFIs address newer and faster I/O interconnects, e.g., 100Gb/s and beyond
- Produce test methodologies and suites to facilitate resilience analysis of applications and systems

Potential End Users:

- DOE/DOD/other government agencies
- HPC researchers and vendors
- Science application developers

Point of Contact: Nathan DeBardleben, HPC, 505-412-1069, ndebard@lanl.gov